JSC-10140
28 September 1976

CR 151110

Field Spectrometer (S191H) Preprocessor Tape Quality Test Program Design Document

Contract NAS 9-15014

(NASA-CR-151110) FIELD SPECTROMETER (S191H) PREPROCESSOR TAPE QUALITY TEST PROGRAM DESIGN DOCUMENT (Aeronutronic Ford Corp.) 595 p HC A25/MF A01 CSCL 14B

N77-12364

Unclas

G3/35 55808

PREPARED FOR

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

LYNDON B. JOHNSON SPACE CENTER

HOUSTON, TEXAS

Aeronutronic Ford

Aeronutronic Ford Corporation
Space Information Systems Operation

1002 Gemini Avenue
Houston Texas 77058

Doc. Type 2
Data Cat. DM

JSC-10140
28 September 1976

FIELD SPECTROMETER (S191H)
PREPROCESSOR TAPE QUALITY TEST
PROGRAM DESIGN DOCUMENT

Contract NAS 9-15014

DRL LI No. 19

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

Submitted by: H. M. Campbell
H. M. Campbell, Supervisor
Support Systems Section

Approved by: H G Johnson
H. G. Johnson, Manager
Software Systems Department

W. A. Danley
W. A. Danley, Manager
Software Product Assurance

Prepared by

AERONUTRONIC FORD CORPORATION
SPACE INFORMATION SYSTEMS OPERATION
1002 GEMINI AVENUE
HOUSTON, TEXAS

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES (CONTINUED)

ACRONYM LIST

ANPRO	-	Subroutine that processes anomalies.
BIAPRO	-	Subroutine that processes bias voltage data during a specified time interval.
CCT	-	Computer compatible tape.
CONINP	-	Control Input Processor.
DCOM2N	-	Decommutation Processor Step 2 transfers data from disk or 9-track tape to core.
ERRPRC	-	Error Processor.
EU	-	Engineering Units
FLDPRO	-	Subroutine that performs limit checks on housekeeping parameters.
QASUM	-	Subroutine used to print a summary of the data derived during the processing.
RAMPRO	-	Subroutine that processes wavelength ramp data during a calibration period.
RESPRO	-	Subroutine used to compute the responsivity.
TMLOOP	-	Subroutine that generates production index arrays and numerical pointers for processing periods.
WVLPRO	-	Subroutine used to process wavelength cal data.

1.0 PROGRAM ABSTRACT AND HISTORY OF USE

1.1 ABSTRACT

Program QA191H performs quality assurance tests on Field Spectrometer data recorded on 9-track magnetic tape. The quality testing involves the comparison of key housekeeping and data parameters with historic and predetermined tolerance limits. Samples of key parameters are processed during the Calibration period and Wavelength Cal period, and the results are printed out and recorded on a historical file tape.

1.2 HISTORY OF USE

The Field Spectrometer Quality Testing Program Design Document is adapted from the S191 Infrared Spectrometer Program Design Document (ERS-300-03) dated February 1973.

2.0 PROGRAM USER'S INFORMATION

2.1 IDENTIFICATION

Title:	QA191H
Author:	W. Ortolani, D. Starbuck
Date:	August 1976
Installation:	MSC, Houston, Texas
Authorization:	Contract Number NAS 9-15014
Source Lanuage:	FORTRAN IV
Computer Configuration:	Digital Equipment Corporation PDP 11/45 DEC CR11 Punched Card Reader DEC RK05 Disk Drive DEC LP11 Line Printer DEC LA30 DECwriter BUCODE Mag Tape Drive (4 each)
Operating System:	DOS 8

2.2 DESCRIPTION OF PROGRAM REQUEST

Reference Earth Resources Production Processing Requirements for EOAP Electronic Sensors, PHO-523, Section 17, Field Spectrometer.

2.3 USAGE

2.3.1 Operational Procedures

QA191H is the resident program to be executed using a Preprocessor (CCT) 9-track tape as input and producing output products. A data flow showing inputs and outputs for QA191H is shown in Figure 2.3.1-1.

A typical run deck setup is shown in Figure 2.3.1-2. The lead card setups are shown in Table 2.3.1-1. The lead card setup shown in Table 2.3.1-1 is used to generate all outputs.

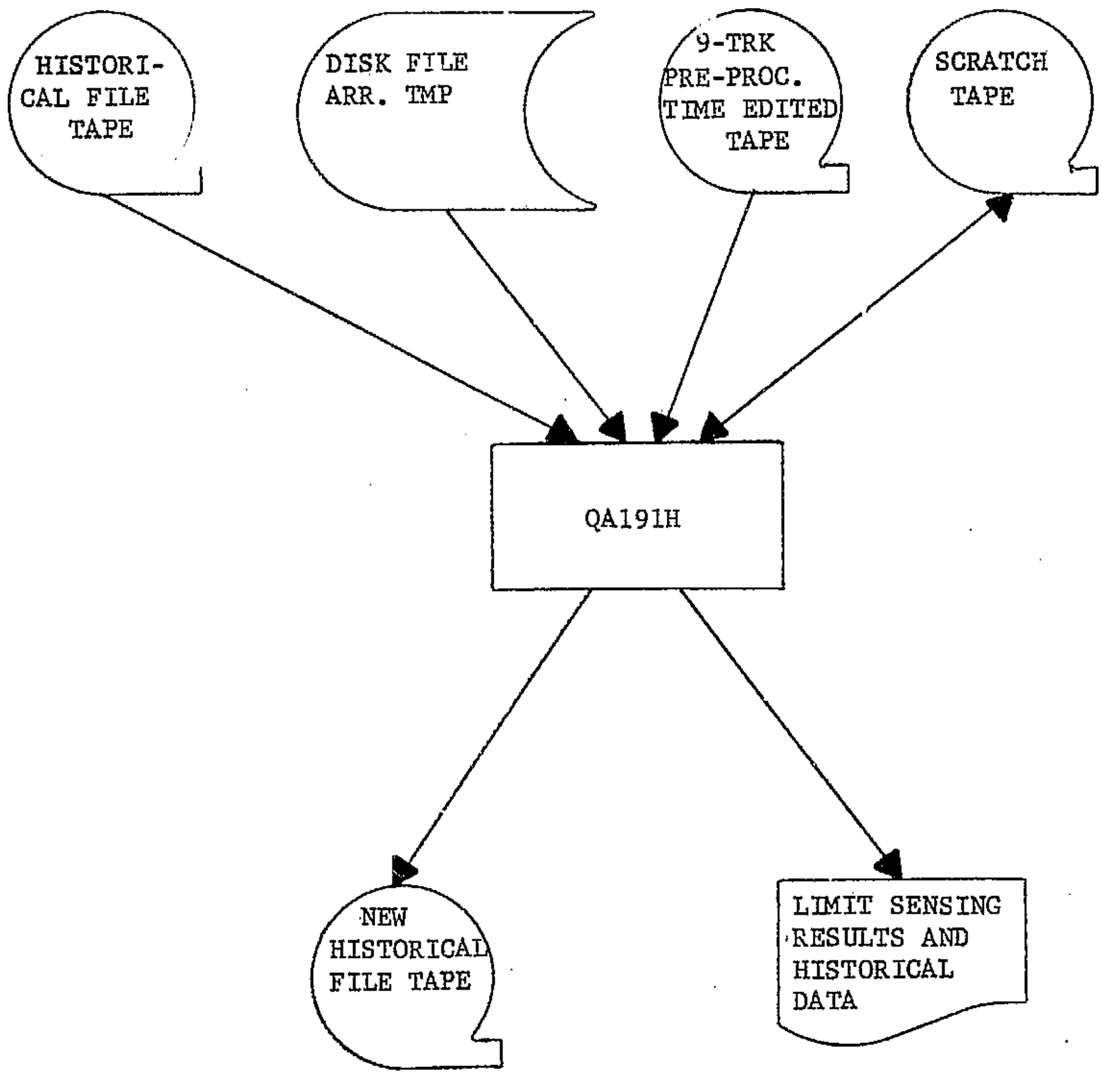
Disk requirements for QA191H are the QA191H system disk on DK0 and the production utility scratch disk on DK1.

The data input tape is in the PDP Format as described in Section 9 of TR543, Volume 2, Earth Resources Data Format Control Book.

The following events will occur during execution:

- Lead cards read from card reader.
- Lead card images and control input parameters printed on line printer.
- Data being read from MT0.
- Output limit sensing results to line printer.
- Historical data being read from MT1.
- New and old historical data output to line printer.
- New and old historical data output to MTAØ.

The outputs to be expected from QA191H are described in Appendices A, B, and C.



QA191H PRODUCT
DATA FLOW

FIGURE 2.3.1-1

```
$JOB QA191H[200,200]
$ME MOUNT QA191H PROD. DISK ON DKO
$ME MOUNT PRODUCTION UTILITY SCRATCH DISK ON DK1
$ME MOUNT PRE-PROCESSOR INPUT TAPE ON MT0; NO RING
$ME MOUNT HISTORICAL INPUT TAPE ON MT1; NO RING
$ME MOUNT HISTORICAL OUTPUT TAPE ON MTA0; WITH RING
$ME MOUNT SCRATCH TAPE ON MTA1; WITH RING
$WAIT
$AS MT0:,9
$AS MT1:,10
$AS MTA0:,11
$AS MTA1:,12
$RUN PIP
#DK0:ARR.TMP[200,200]/UN
#DK0:ARR.TMP[200,200]/DE
#MTA0:/ZE
#MTA1:/ZE
$RUN FLOAD.LDA[100,100]
#/EXIT
$RUN DK0:QA191H.LDA[200,200]
LEAD CARDS
(As shown in Table 2.3.1-1)
$FI
```

Figure 2.3.1-2 QA191H Run Deck Setup

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
01	ISENSR	1-5	I5	15	SENSOR CODE
	IRFMT	6-10	I5	27	RECORDING FORMAT
	IMIS	11-15	I5	0-32767	MISSION NUMBER (NEGATIVE = DEBUG)
	IFLT	16-20	I5	0-32767	FLIGHT NUMBER
	ISIT	21-25	I5	0-32767	SITE NUMBER
	ILIN	26-30	I5	0-32767	LINE NUMBER
	IRUN	31-35	I5	0-32767	RUN NUMBER
02	IPROJ	1-4	2A2	-----	PROJECT IDENTIFICATION
	ITITLE	6-65	30A2	-----	TITLE INFORMATION
03	IYR	1-5	I5	50-99	BASE YEAR OF DATA RECORDING DATE
	IMON	6-10	I5	1-12	BASE MONTH OF DATA RECORDING DATE
	IDAY	11-15	I5	1-31	BASE DAY OF DATA RECORDING DATE
	IHTAB	16-20	I5	0-2	0=TAB OUT NEW HISTORICAL FILE.
					1=TAB OUT OLD AND NEW HISTORICAL FILE.
					2=TAB OUT OLD HISTORICAL FILE ONLY;
					NO OTHER PROCESSING
	IDELOP	21-25	I5	0-1	0-FILE DELETION IS NOT REQUIRED
					1-FILE DELETION IS REQUIRED
	IDELHR	26-30	I5		TIME OF FILE TO BE DELETED IN HOURS,
	IDELMN	31-35	I5		MINUTES, AND SECONDS. IF IDELOP IS 0,
	SDELSC	36-40	I5		COLUMNS 26-40 ARE LEFT BLANK.

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
04	NOVRL	1-5	I5	1-20	NUMBER OF OVERALL TIME PERIODS TO PROCESS
	NCALP	6-10	I5	0-5	NUMBER OF CAL PERIODS TO BE PROCESSED
	NWCAL	11-15	I5	0-5	NUMBER OF WAVELENGTH CAL PERIODS TO PROCESS
	NTOTAL	16-20	I5	10-65	TOTAL NUMBER OF TIME PERIODS TO BE PROCESSED (NOVRL + (NCALP (NOVRL + (NCALP*7) + (NWCAL*2))
*05	ISTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	ISTM	6-10	I5		SECONDS FOR INTERVAL OF TAPE TO
	STS	11-20	F10.0		PROCESS
	ISPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	ISPM	26-30	I5		SECONDS FOR INTERVAL OF TAPE TO
	SPS	31-40	F10.0		PROCESS

*Card type 5 must be entered NOVRL times. NOVRL can equal 1-20. All overall times must be entered in chronological order.

Table 2.3.1-1Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
*06	ICSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	ICSTM	6-10	I5		SECONDS OF THE CALIBRATION PERIOD(S)
	STCS	11-20	F10.0		TO PROCESS
	ICSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	ICSPM	26-30	I5		SECONDS OF THE CALIBRATION PERIOD(S)
	SPCS	31-40	F10.0		TO PROCESS
**07	IWSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IWSTM	6-10	I5		SECONDS OF THE OVERALL WAVELENGTH
	STWS	11-20	F10.0		CAL PERIOD(S).
	IWSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	IWSPM	26-30	I5		SECONDS OF THE OVERALL WAVELENGTH
	SPWS	31-40	F10.0		CAL PERIOD(S).
***08	IBSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IBSTM	6-10	I5		SECONDS OF THE BIAS VOLTAGE
	STBS	11-20	F10.0		PROCESSING PERIOD
	IBSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	ISBPM	26-30	I5		SECONDS OF THE BIAS VOLTAGE
	SPBS	31-40	F10.0		PROCESSING PERIOD

*Card type 6 entered NCALP times.
**Card type 7 entered NWCAL times.

***Card type 8 is entered NCALP times, one card for each cal period requested.

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
*09	IHCSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IHCSTM	6-10	I5		SECONDS OF THE HEATED CAL LIMITS
	STHCS	11-20	F10.0		PERIOD FOR RADIANCE CAL WHEEL
					POSITION CHECK
	IHCSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	IHCSPM	26-30	I5		SECONDS OF THE HEATED CAL LIMITS
	SPHCS	31-40	F10.0		PERIOD FOR RADIANCE CAL WHEEL
					POSITION CHECK
*10	ISCSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	ISCSTM	6-10	I5		SECONDS OF THE SWL CAL LIMITS PERIOD
	STSCS	11-20	F10.0		FOR RADIANCE CAL WHEEL POSITION CHECK
	ISCSTH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	ISCSTM	26-30	I5		SECONDS OF THE SWL CAL LIMITS PERIOD
	SPSCS	31-40	F10.0		FOR RADIANCE CAL WHEEL POSITION CHECK
**11	IACSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IACSTM	6-10	I5		SECONDS OF AMBIENT CAL LIMITS PERIOD
	STACS	11-20	F10.0		FOR RADIANCE CAL WHEEL POSITION CHECK
	IACSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	IACSPM	26-30	I5		SECONDS OF AMBIENT CAL LIMITS PERIOD
	SPACS	31-40	F10.0		FOR RADIANCE CAL WHEEL POSITION CHECK

*Card types 9 & 10 are entered NCALP times, one card for each cal period requested.

**Card type 11 entered NCALP times, one card for each cal period requested.

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
*12	IHRSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IHRSTM	6-10	I5		SECONDS OF HEATED CAL RESPONSIVITIES
	STHRS	11-20	F10.0		PROCESSING PERIOD
	IHRSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	IHRSPM	26-30	I5		SECONDS OF HEATED CAL RESPONSIVITIES
	SPHRS	31-40	F10.0		PROCESSING PERIOD
*13	ISRSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	ISRSTM	6-10	I5		SECONDS OF SWL CAL RESPONSIVITIES
	STSRS	11-20	F10.0		PROCESSING PERIOD
	ISRSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	ISRSPM	26-30	I5		SECONDS OF SWL CAL RESPONSIVITIES
	SPSRS	31-40	F10.0		PROCESSING PERIOD
**14	IWLSTH	1-5	I5		START TIME IN HOURS, MINUTES, AND
	IWLSTM	6-10	I5		SECONDS FOR THE WAVELENGTH CAL
	STWLS	11-20	F10.0		PROCESSING PERIOD
	IWLSPH	21-25	I5		STOP TIME IN HOURS, MINUTES, AND
	IWLSPM	26-30	I5		SECONDS FOR THE WAVELENGTH CAL
	SPWLS	31-40	F10.0		PROCESSING PERIOD
15	ICSCN	1-5	I5	1-100	NUMBER OF CONSECUTIVE CAL PERIOD
					SCANS TO PROCESS FOR WAVELENGTH RAMP
					CALCULATIONS
	ILTOL	6-10	I5		THE PLUS OR MINUS TOLERANCE LIMIT FOR
					PCM COUNT DEVIATION DURING LINEARITY
					OF SCAN CHECK

*Card types 12 & 13 are entered NCALP times, one card for each cal period requested.

**Card type 14 entered NWCAL times.

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
CONT. 15	ISYNC	11-15	I5		MINIMUM PCM COUNT VALUE OF PARAMETERS
					A1, A2, A3, A5 and A6 FOR VALID SYNC
					PULSES
	IEND	16-20	I5		MINIMUM PCM COUNT VALUE OF A4 FOR
					VALID END OF SCAN
	ISMIN	21-25	I5		MINIMUM AND MAXIMUM NUMBER OF DATA
	ISMAX	26-30	I5		SAMPLES FOR VALID SCAN
16	IBVHR(1)	1-35	7I5		ASSIGNED VALUE RANGES FOR THE BIAS
	IBVHR(7)				VOLTAGE HISTOGRAMS. FIVE CARDS ARE
					READ, ONE FOR EACH PARAMETER
	IBVHR (29)				(1ST CARD=A1, 2ND CARD=A2, 3RD CARD =
	IBVHR (35)				A3, 4TH CARD=A5, 5TH CARD=A6)
*17	WVLNGH	1-10	F10.0		WAVELENGTH (λ) VALUE FOR EACH PARAMETER
					TO BE USED IN RESPONSIVITY COMPUTATION
	PDORL	11-20	F10.0		CAL SOURCE BRIGHTNESS (L) FOR PARA-
					METERS A2, A3, and A5. VALUE FOR
					DICHROIC REFLECTIVITY (Pd) FOR PARA-
					METER A6.
	CBIASV	21-30	F10.0		CHANNEL BIAS VOLTAGE FOR EACH PARA-
					METER
	ACOEFZ	31-40	F10.0		RESPONSIVITY POLYNOMIAL COEFFICIENT
	ACOEF1	41-50	F10.0		VALUES FOR EACH PARAMETER, FOR
	ACOEF2	51-60	F10.0		EXPANDING WAVELENGTH TO A4 COUNTS.
	ACOEF3	61-70	F10.0		
18	IWLLO	1-5	I5		LOW PCM COUNT LIMIT OF A4 FOR WAVE-
					LENGTH CAL PARABOLA MATRIX.
	IWLHI	6-10	I5		HIGH PCM COUNT LIMIT OF A4 FOR WA[illegible]
					LENGTH CAL PARABOLA MATRIX

*Card type 17 entered 4 times, one card for each parameter . First Card =A2, Second Card = A3, Third Card = A5, Fourth Card = A6.

Table 2.3.1-1 Lead Card Setup for QA191H

Card Type	Mnemonic	Columns	Format	Range	Description
*19	IHMAX(I)	1-10	F10.0		MAXIMUM AND MINIMUM TOLERANCE LIMITS, IN PCM COUNTS, FOR PARAMETER QUALITY CHECK. FOUR PAIRS OF LIMITS PER CARD, IN THE FOLLOWING ORDER: 1=ZERO VOLTS REF.;2=POWER SUPPLY DIAG;3=PACKAGE TEMP;4=DATA PALLET TEMP;5=SPECTROMETER PALLET TEMP;6=DICHROIC TEMP;7=INT. SPHERE TEMP;8=MIRROW TEMP;9=HEATED CAL TEMP;10=AMBIENT CAL TEMP;11=LWL DETECTOR TEMP;12=THERMAL REF TEMP; 13=SWL CAL LAMP.
20	IRMAX(I)	1-5	I5.		MAXIMUM AND MINIMUM TOLERANCE LIMITS, IN PCM COUNTS, FOR QUALITY CHECK OF RADIANCE CAL WHEEL POSITION. FOUR PAIRS OF LIMITS IN THE FOLLOWING ORDER: 1=FIELD DATA LIMITS; 2=HEATED CAL LIMITS; 3=SWL CAL LIMITS; 4=AMBIENT CAL LIMITS.
	IRMIN(I)	6-10	I5.		
	I = 1, 4				
21	IFMAX(I)	1-5	I5		MAXIMUM AND MINIMUM TOLERANCE LIMITS IN PCM COUNTS, FOR QUALITY CHECK OF FOV FLAG. THREE PAIRS OF LIMITS IN THE FOLLOWING ORDER: 1=BOTH CHANNELS ACTIVE; 2=SWL CHANNEL ACTIVE; 3=LWL CHANNEL ACTIVE.
	IFMIN(I)	6-10	I5		
	I = 1, 3				

* 2 cards: 8 values in first card; 5 values in second.

2.3.2 Hardware Configuration Required

- DEC Writer
- Card Reader
- Line printer
- Tape drives (4)
- Disk drives (1)

The normal tape drive to logical unit assignments are:

```
$AS MT0:,9
$AS MT1:,10
$AS MTA0:,11
$AS MTA1:,12
```

These assignments of tape drive to logical unit may be switched if desired by changing the assign cards. If this is done, care should be exercised to see that the correct tapes are mounted on the tape drives.

2.3.3 Restrictions

A. The maximum number of overall time intervals requested may not exceed 20 for any one run.

B. The maximum number of Calibration periods and Wavelength Cal periods requested may not exceed 5 (each) for any one run.

2.3.4 Error Diagnostics

Appendix C lists the error diagnostics which may be printed by the Production Processing System. QA191H error messages are included in this list.

2.3.5 Core/Disk Requirements

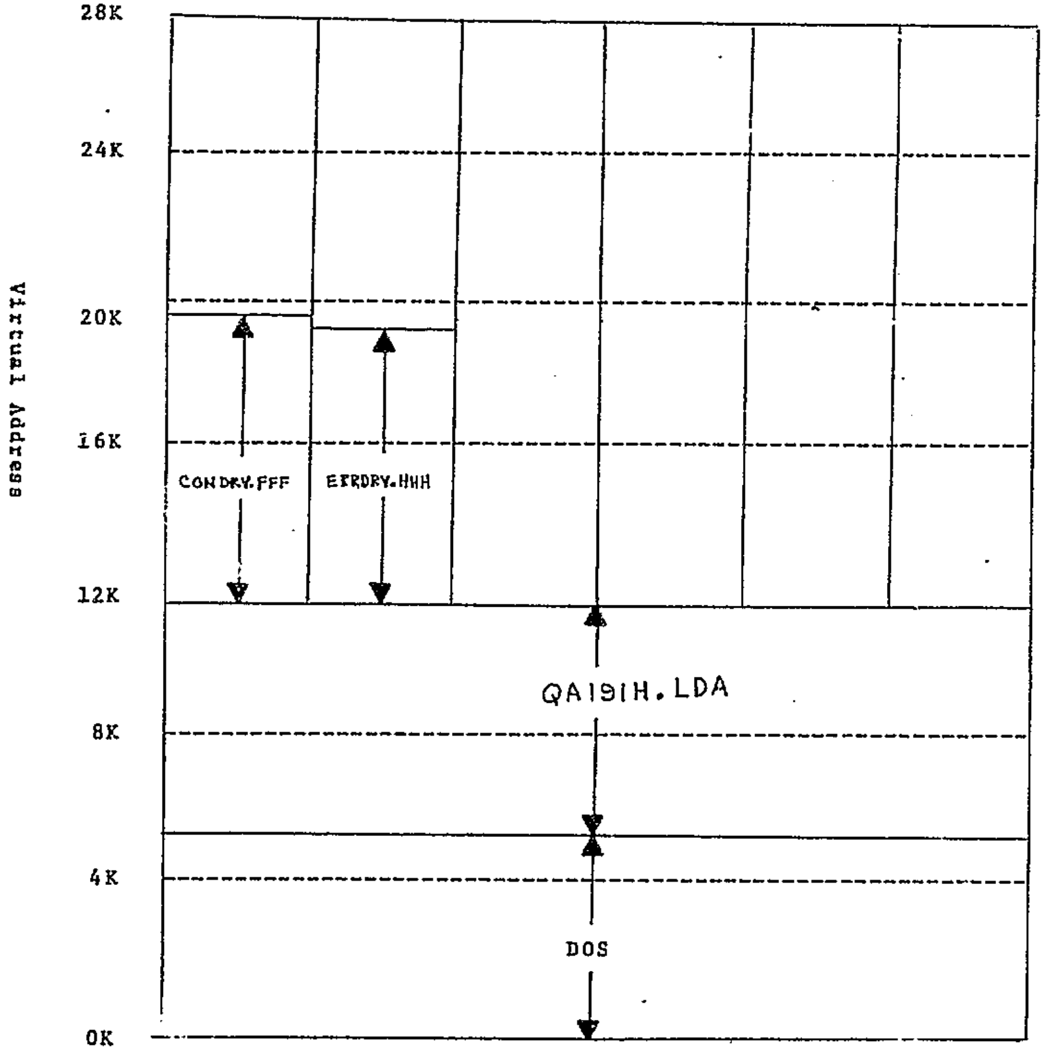
A. See Figures 2.3.5-1 and 2.3.5-2 for virtual and physical core requirements.

B. The temporary storage file ARR.TMP, requires 4325 words of disk space.

Logical Unit No	Device	Filename	Length (If Applicable)	Description
2	DK0:	ARR.TMP	3425 WORDS	TEMPORARY STORAGE FOR
				HISTORICAL DATA RECORDS
5	CR:			CARD READER
6	LP:			LINE PRINTER
9	MT0:			INPUT TAPE DRIVE
10	MT1:			INPUT TAPE DRIVE
11	MTA0:			OUTPUT TAPE DRIVE
12	MTA1:			SCRATCH TAPE DRIVE

Table 2.3.2-1 Standard Logical Unit Assignments for QA191H

Allocation of Addressable Core for QA191H



Physical Core Requirements
for
Memory Mapping Configuration AAA

DOS	5.2K
Resident Program	
QA191H.LDA	6.8K
Load Modules	
CONDRV.FFF	7.7K
ERRDRV.HHH	7.2K

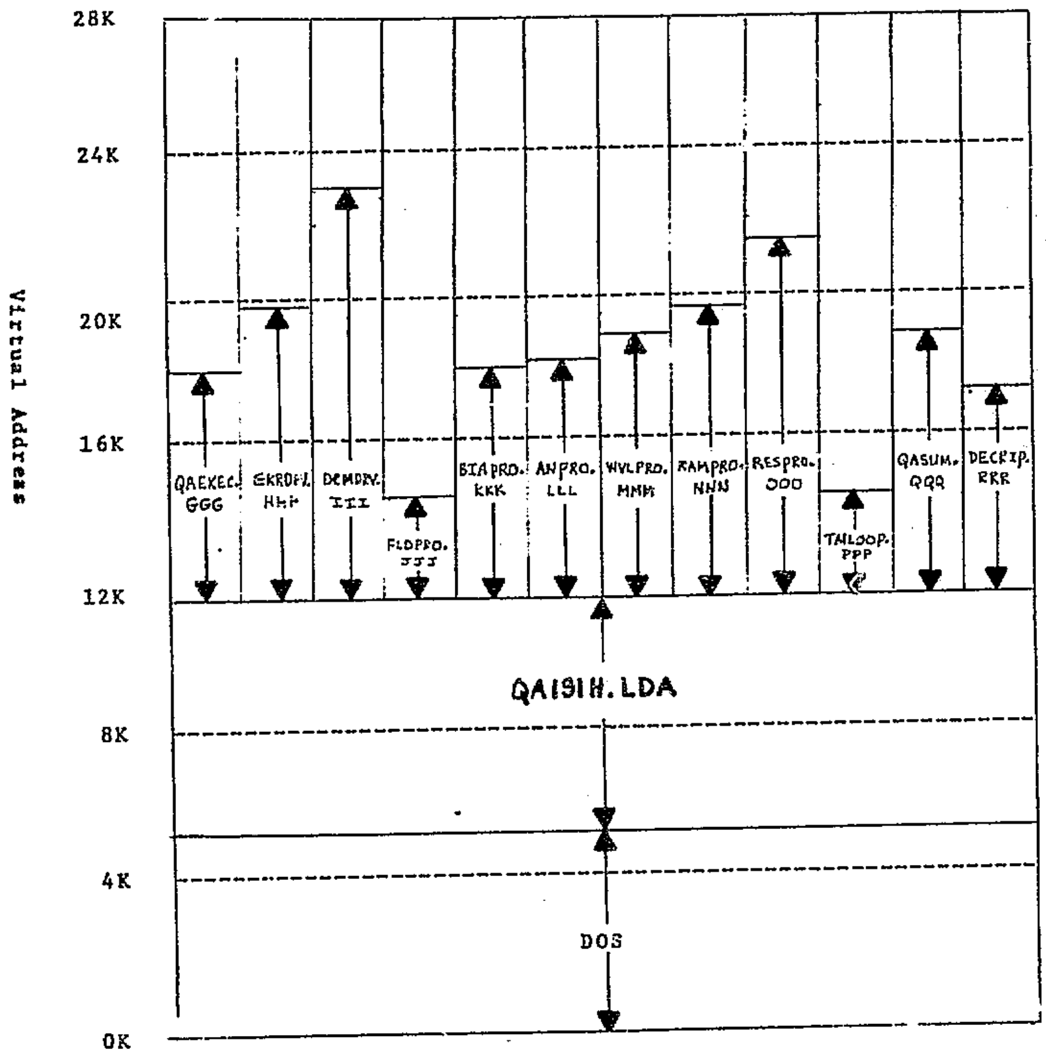
Data Areas

Name	Size	No. Alloc.	
DCMBUF.DA1	4K	1	3.9K

SEG	4K
Total Physical Core	34.8K

Figure 2.3.5-1 Allocation of Addressable & Physical Core for QA191H

Allocation of Addressable Core for QA191H



Physical Core Requirements
for
Memory Mapping Configuration BBB

DOS	5.2K
Resident Program QA191H.LDA	6.8K
Load Modules	
QAEXEC.GGG	5.7K
ERRDFL.HHH	7.2K
DCMDRV. III	11.1K
FLDPRO. JJJ	2.4K
BIAPRO. KKK	5.7K
ANPRO. LLL	6.0K
WVLPRO. MMM	6.75K
RAMPRO. NNN	7.4K
RESPRO. OOO	9.75K
TMLOOP. PPP	2.4K
QASUM. QQQ	10.6K
DECRIP. RRR	9.1K

Data Areas Name	Size	No. Alloc.	
DCMBUF.DA1	4K	1	3.9K
SEG			4K
Total Physical Core			103.5K

Figure 2.3.5-2 Allocation of Addressable & Physical Core For QA191H

2.4 ANALYSIS

2.4.1 Description

QA191H is the core resident program which controls the data processing by testing for options and times and calling the appropriate programs. Program control information is read from lead cards by calling CONDRV. Control information is used for data identification, tolerance limits, processing options, tabulation option, and file deletion options. There are four primary processing modes. One is for field data housekeeping parameters, a second is for Calibration period data, a third is for wavelength cal period data, and the fourth is for building a new Historical Data File tape. The processing modes are as follows:

A. The field data housekeeping parameters are processed to test actual data values against nominal tolerance limits.

   Subroutine DECRIP is called to read the descriptor file from the S191H preprocessor 9-track tape and store the header information for DCOM2N.

   Subroutine DCOM2N is called to read the data from the S191H preprocessor 9-track tape and store it in a common array.

   Subroutine FLDPRO is called to initiate housekeeping parameter checks. The tolerance limits, in PCM counts, are initiated for those parameters being tested and an algorithm is used to determine the location of the start of the first whole scan of data in the MEAS array. Each parameter data sample is then compared with its associated set of limits.

   If any of the sample values exceed the tolerance limits, subroutine ANPRO is called to process the anomaly. ANPRO first checks to determine if the particular parameter on which the anomaly occurred also had an anomalous value on the preceding sample. The first time two consecutive sample anomalies are detected for a given parameter, a flag is set and a counter is initiated. Thereafter, for a predetermined number of scans, the counter keeps a running total of all anomalies that occur on that parameter. When the predetermined number of scans have been processed or the time between frames has equalled or exceeded 30 seconds, the anomaly total, the number of scans, and parameter identification is output to the line printer. If the anomaly is the first of a series or the only one detected, the PCM count value of the sample is converted to appropriate engineering values and the time parameter and value is output to the line printer.

B. If Calibration period data is to be processed, subroutine RAMPRO is called to compute wavelength ramp data. For a predetermined number of consecutive valid scans, the peak and minimum PCM count values for channel A4 are determined. Those A4 values for the total number of scans are used to compute the average and standard deviation values. An algorithm

2.4.1 B. Cont.

is used to compute a nominal straight line value for the channel A4 ramp of each scan. The actual A4 PCM values are compared with the computed straight line values, sample for sample, and if the actual values fall within a nominal tolerance limit, the RMS deviation is computed for all sample values. If a value exceeds the straight line limits, all samples for that scan are rejected, the A4 PCM count of the anomalous sample is converted to volts, and the data is output to the line printer with an identifying label and time. The length of time of each valid scan is computed and stored in order that the average length of time for the group of consecutive valid scans may be determined. The sync pulses are also tested at the start of each scan. When a start of scan is detected, the first data sample of all channels except A4 are checked for saturation (PCM count 1020). If any of the five channels do not satisfy the saturation criteria, the frame time, channel number and value, and anomaly description are output to the line printer. The computed values for A4 peak and minimum voltage, RMS deviation, and average scan length are stored for building the Historical Data file.

Subroutine BIAPRO is called to process bias voltage data when the control card input times are satisfied. Histograms of the bias voltages on channels A1, A2, A3, A5, and A6 are generated from 2 seconds (approximately 2 scans) of data. Arrays are initiated for the five channels, and the PCM count values of the channel data samples are enumerated within seven specified value ranges. The PCM values for the sync pulses and for the four data samples on each side of the sync pulses are not used in the histograming process. The Histograms for each of the five channels are stored for building the Historical Data file.

The calibration period data on the preprocessor 9-track tape represents three different phases in the calibration cycle; the ambient cal period, the heated cal period, and the SWL cal period. The representative times for these three periods are input to the processor by control cards.

During the SWL cal period PCM count tolerance limits are initiated for the SWL Cal lamp and Radiance Cal Wheel position checks. The PCM value of each SWL Cal lamp data sample is tested against the tolerance limits. If an actual value exceeds the limits, the PCM value is converted to volts, and the anomaly data is output to the line printer. The SWL Cal lamp PCM values are tested only during the SWL Cal period.

2.4.1 B. Cont.

Similarly, during the heated cal and ambient cal periods, different PCM count tolerance limits are initiated for the Radiance Cal Wheel position checks.

Responsivities will be derived from one wavelength in each of channels A2, A3, A5, and A6 using 20 consecutive valid scans of calibration period data. Channels A2, A3, and A5 are derived from the SWL cal period data, and channel A6 is derived from the heated cal period data. No responsivities are processed from the ambient cal period data. Subroutine RESPRO is called to compute the responsivities. A polynomial algorithm using a coefficient table converts the requested wavelengths to a representative channel A4 PCM count. The channel A4 data drift correction is inserted by multiplying the PCM count by the ratio of the peak value of A4 for the given scan over the nominal A4 peak value. The actual A4 PCM count closest to the computed value is determined for each of the 20 scans. The actual PCM count is then averaged with the two preceding and two following sample counts. The 20 computed values are then converted to volts, and the average and standard deviation values are obtained. Channel bias voltages are then subtracted from the averages to obtain the signal volts. The responsivity for channels A2, A3, and A5 are derived by dividing the signal volts by the wavelength brightness of the associated wavelength. The wavelength and wavelength brightness values are input through control cards, as are the channel bias voltages. To obtain the responsivity for channel A6, the reference source, heated cal source, and dichroic temperatures are derived for that cal period. These temperatures are used to calculate Heated source radiance and Reference source radiance. Responsivity is then obtained by dividing the channel A6 signal volts by the Heated source radiance minus the Reference source radiance. The noise-equivalent spectral radiance (NESR) for each channel is obtained by dividing its standard deviation value (Noise) by its responsivity value. The responsivity, noise, and NESR data is then stored for building the Historical Data file.

All the housekeeping parameters that are tested during field data processing are also tested during the calibration periods. The Ambient Cal Temperature, Heated Cal Temperature, and SWL Cal Lamp parameters are not tested during field data processing. These three parameters are only tested when their associated cal period data is processed. Most of the housekeeping parameters require only one set of PCM count tolerance limits. The following parameters are exceptions:

2.4.1 B. Cont.

1. The Radiance Cal Wheel Position parameter requires four sets of PCM count tolerance limits. A separate set of limits is required while processing field data, heated cal data, SWL cal data, and ambient cal data.

2. The Field of View flag parameter requires three sets of PCM count tolerance limits. A separate set of limits is required for the three conditions; both channels active, LWL channel only active, or SWL channel only active.

3. The LWL Detector Temperature parameter requires only one set of tolerance limits, but when bias voltage is being processed, during the cal periods, the limits are set for saturated data (between 1020 and 1023 counts).

4. One set of initial tolerance limits are required for the Dichroic Temperature and Thermal Reference Source Temperature parameters. The tolerance limits for subsequent processing periods are derived during the calibration period data processing.

All tolerance limit values, except those derived in real time, are input through control cards.

C. Subroutine WVLPRO is called to process wavelength cal data. The first twenty valid scans of wavelength cal data are processed. Predetermined value limits for channel A4 PCM counts are initiated, and arrays are used to build matrices of all values of channels A2, A3, A4, and A5 whose data sample position is within the channel A4 limits. Beginning with channel A3 and ending with A2, the values in each scan of channels A3, A5, or A2 are tested. The channel data is rejected if any values within the A4 range exceed 1000 counts (saturation). The first channel whose twenty scans of data contain no saturation is selected for processing. Within the matrix of the chosen channel, a high and low range of values and their relative A4 values are determined. A least squares regression of the output detector channel values is performed on the A4 values, and the PCM value of A4 at the vertex of the parabola is computed. The computed A4 values are biased by the ratio of the actual A4 peak count over a nominal count value, to compensate for A4 drift. The chosen channel now has 20 computed values of A4, for which average and standard deviation is computed. These values, in PCM counts, are stored for building the Historical File.

D. After all time intervals have been processed, subroutine QASUM is called to print a summary of the data derived during the processing. This summary includes the total number of anomalies found on each housekeeping parameter and flags to indicate any excessively erroneous data areas.

2.4.1 D. Cont.

A new Historical Data file tape is then built. The data from the old Historical Data file tape is read, and the data frame time for each file is checked. If IHTAB equals one, the old historical files are output to the line printer as they are read. The old data frame times are compared with the frame time for the new historical data, and the old data is output to the new Historical Data file tape until the time period for the new data is found. The new historical file is then output to the line printer and the new Historical Data file tape. After the new historical file has been output, and if the E.O.F. on the old historical file tape has not been reached, the remaining files on the old historical tape are read and output to the new Historical Data file tape and line printer (if requested).

By setting IHTAB equal to 2 in the Lead Card Setup, the old Historical File tape can be dumped in its entirety to the line printer. No other processing is performed.

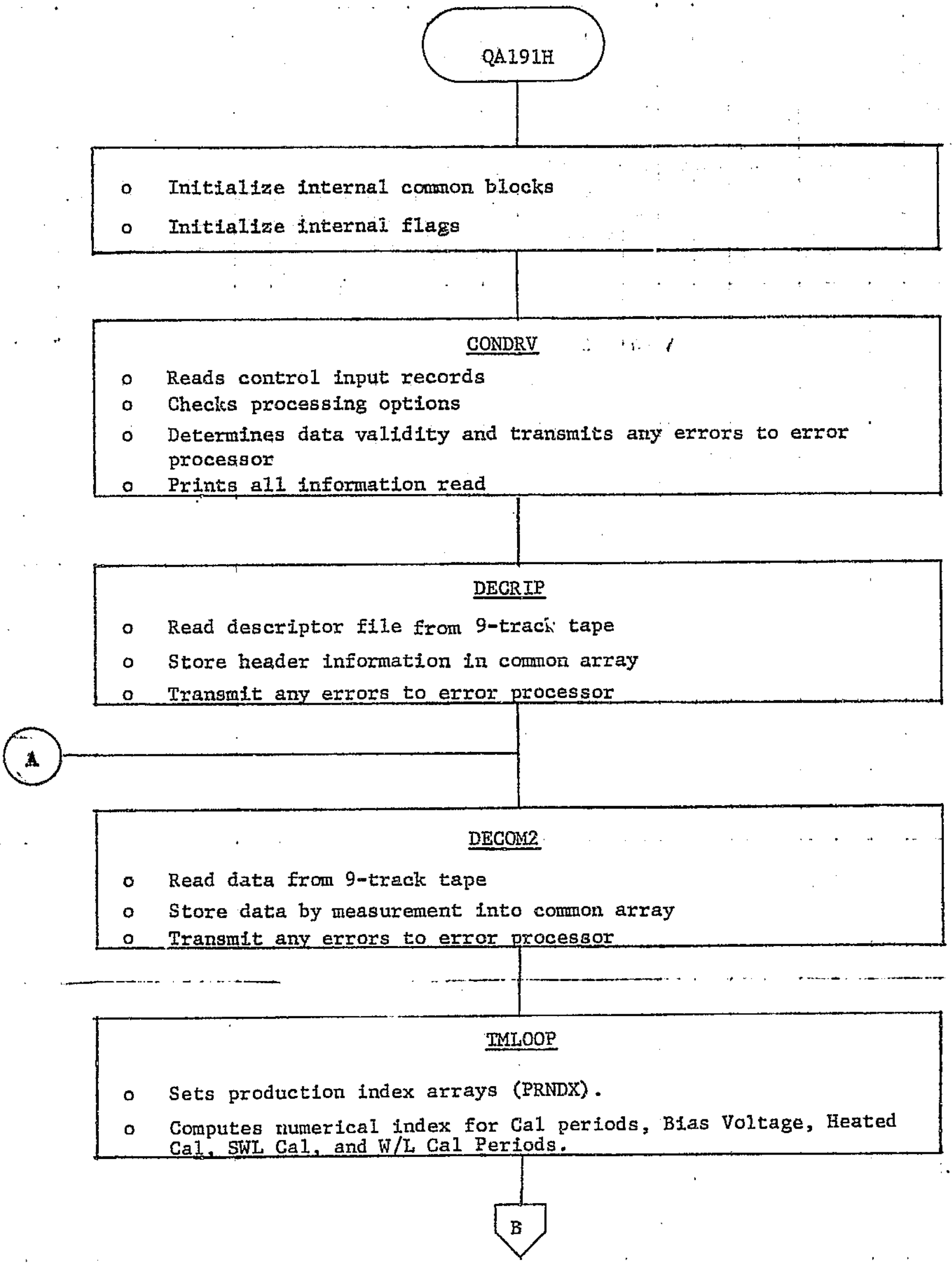


Figure 2.4-1 QA191H DATA FLOW

B

FLDPRO

- o Initializes parameter tolerance limits
- o Checks Field Data parameter values against limits

Any anomaly found ?

Yes → C

No

Process W/L Cal Data ?

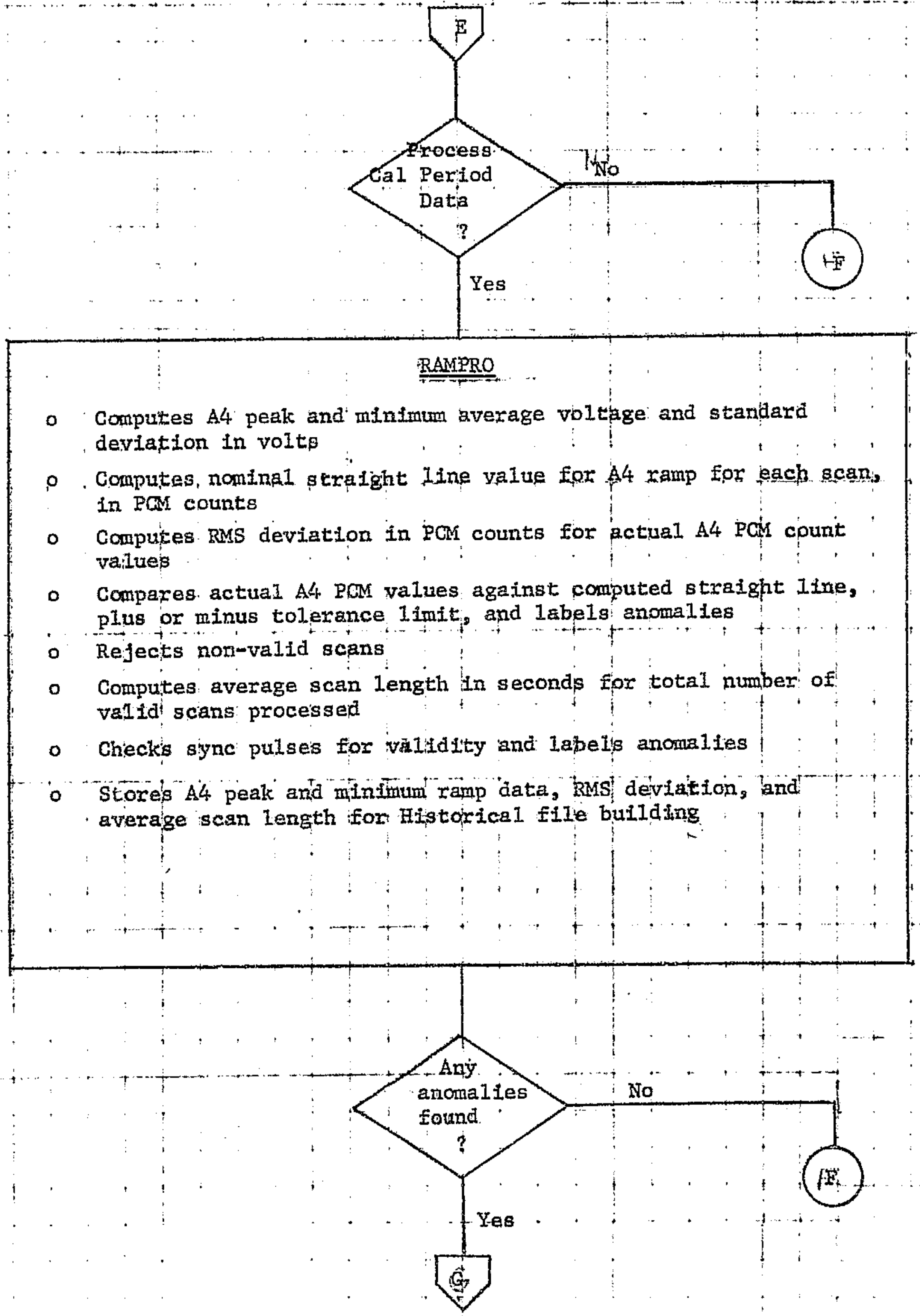
No → E

Yes

WVLPRO

- o Determines data interval of channel A4 samples within pre-selected low and high value limits.
- o Builds matrices of all values of channels A2, A3, A4, and A5 associated with the A4 data interval.
- o Does progressive testing on values of channel A3, A5, and A2, in that order, to detect saturated data.
- o Selects the first channel without saturated data for 20 consecutive scans.
- o Determines first and last occurrenceof the highest PCM value within the matrix of the selected channel.
- o Determines range of values in channel matrix bounded by values equal to one-third the highest value.
- o Computes parabola fit of channel matrix values on channel A4 data, within the range.
- o Determines vertex of each parabola (A4HG) for 20 scans of the selected channel.
- o Compensate for channel A4 drift.
- o Compute average and standard deviation, in PCM counts, for the 20 values of A4HG.
- o Store data for Historical File.

E


E
Process Cal Period Data ?
No
F
Yes
RAMPRO
o Computes A4 peak and minimum average voltage and standard deviation in volts
o Computes nominal straight line value for A4 ramp for each scan, in PCM counts
o Computes RMS deviation in PCM counts for actual A4 PCM count values
o Compares actual A4 PCM values against computed straight line, plus or minus tolerance limit, and labels anomalies
o Rejects non-valid scans
o Computes average scan length in seconds for total number of valid scans processed
o Checks sync pulses for validity and labels anomalies
o Stores A4 peak and minimum ramp data, RMS deviation, and average scan length for Historical file building
Any anomalies found ?
No
F
Yes
G

G

ANPRO

- o A4 straight line anomalies are converted to volts
- o Prints out A4 straight line and sync pulse anomalies

F

Process Responsivity Data ?

NO

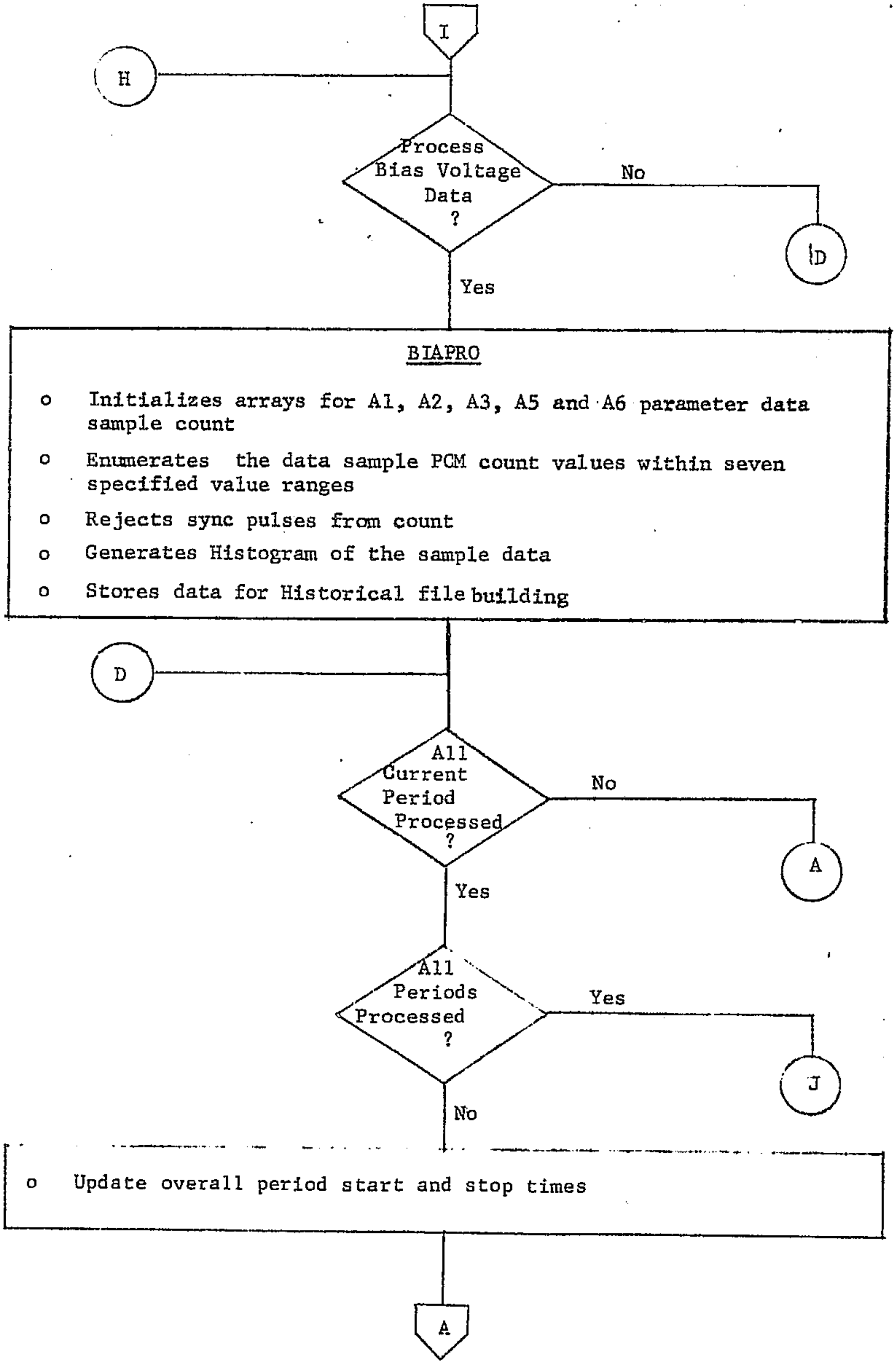
H

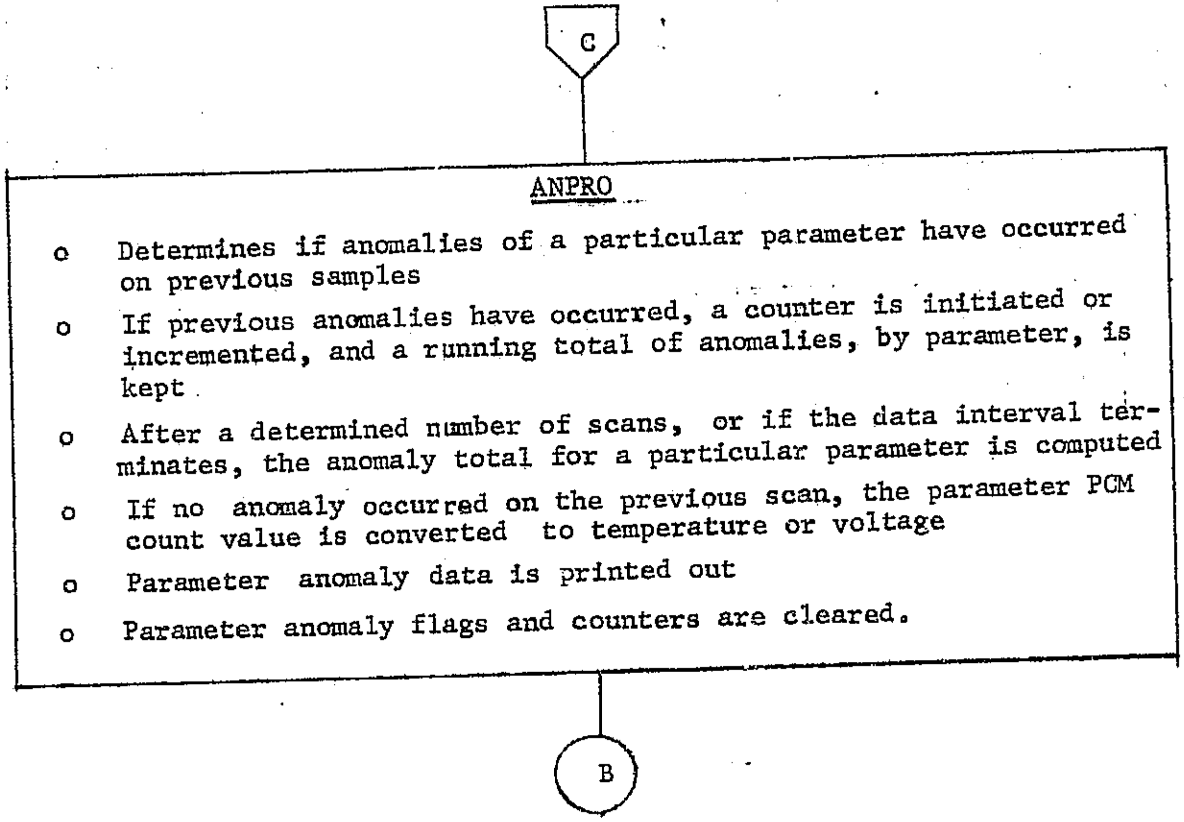
Yes

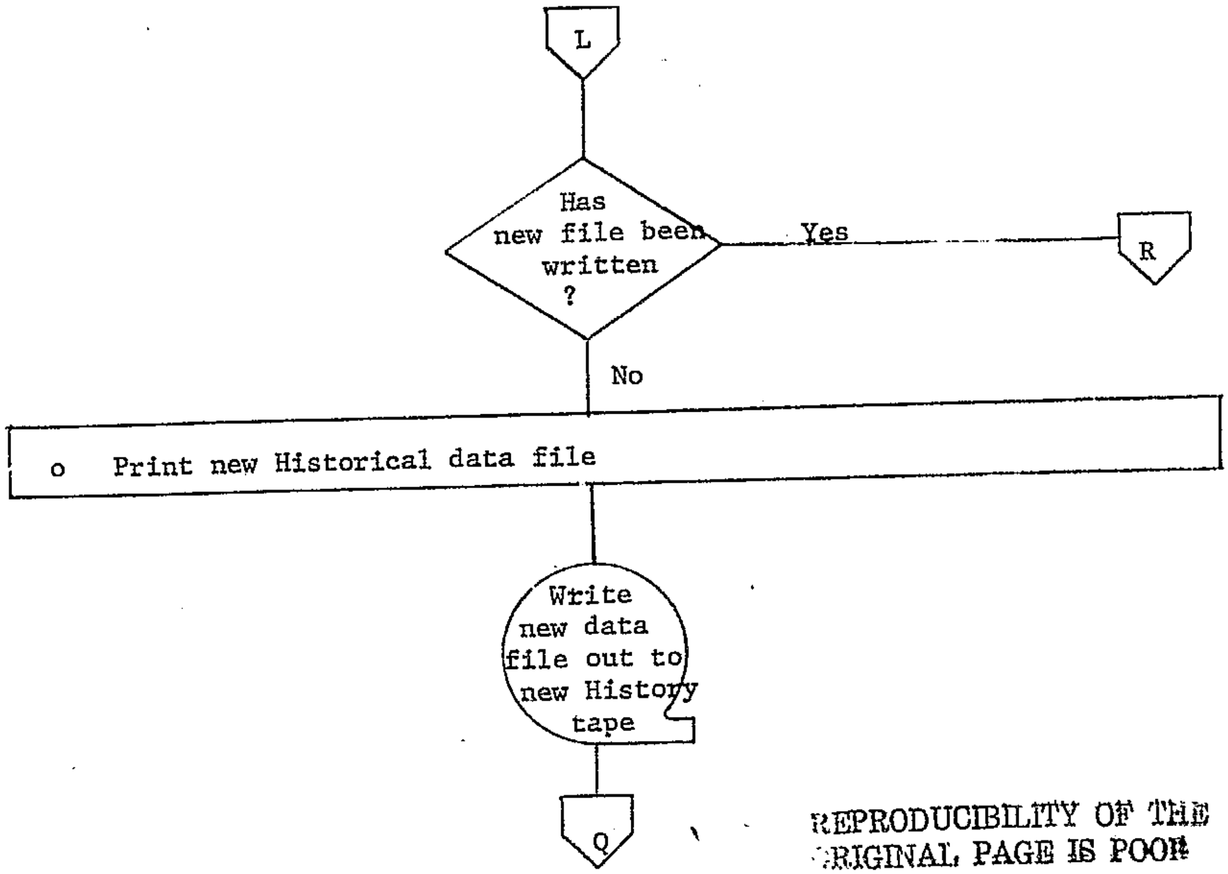
RESPRO

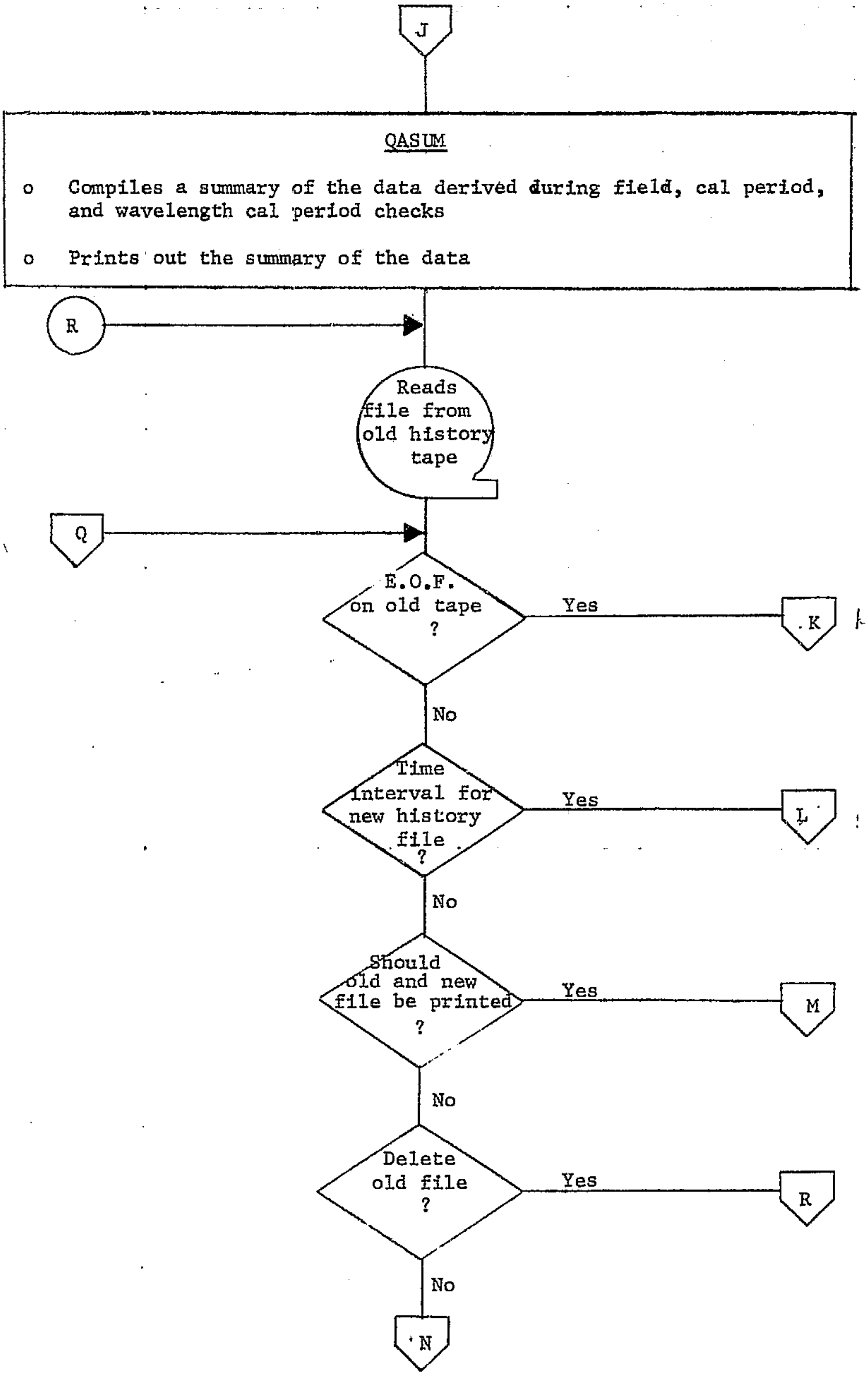
- o Use coefficient table to convert wavelengths to PCM counts.
- o Compensate for channel A4 drift by multiplying the wavelength PCM value by the peak A4 value of the given scan over the nominal peak A4 value (983 counts).
- o Compare new wavelength PCM value with values in channel A4.
- o Determine point where first A4 value equals or exceeds wavelength PCM value.
- o At that point, add the associated output detector value to the two output detector sample values on either side.
- o Find the average of the five output detector values.
- o Convert the average output detector values to volts and store the converted values for 20 consecutive scans.
- o Compute average and standard deviation for the 20 stored values.
- o Insert channel bias voltage correction for the associated output detector.
- o Compute responsivities for channels A2, A3, and A5.
- o Calculate Reference, Heated Cal, and Dichroic temperatures, and use to calculate channel A6 responsivity.
- o Use responsivities and standard deviations to compute NESR for all channels.
- o Store responsivity, noise, and NESR data for Historical File.

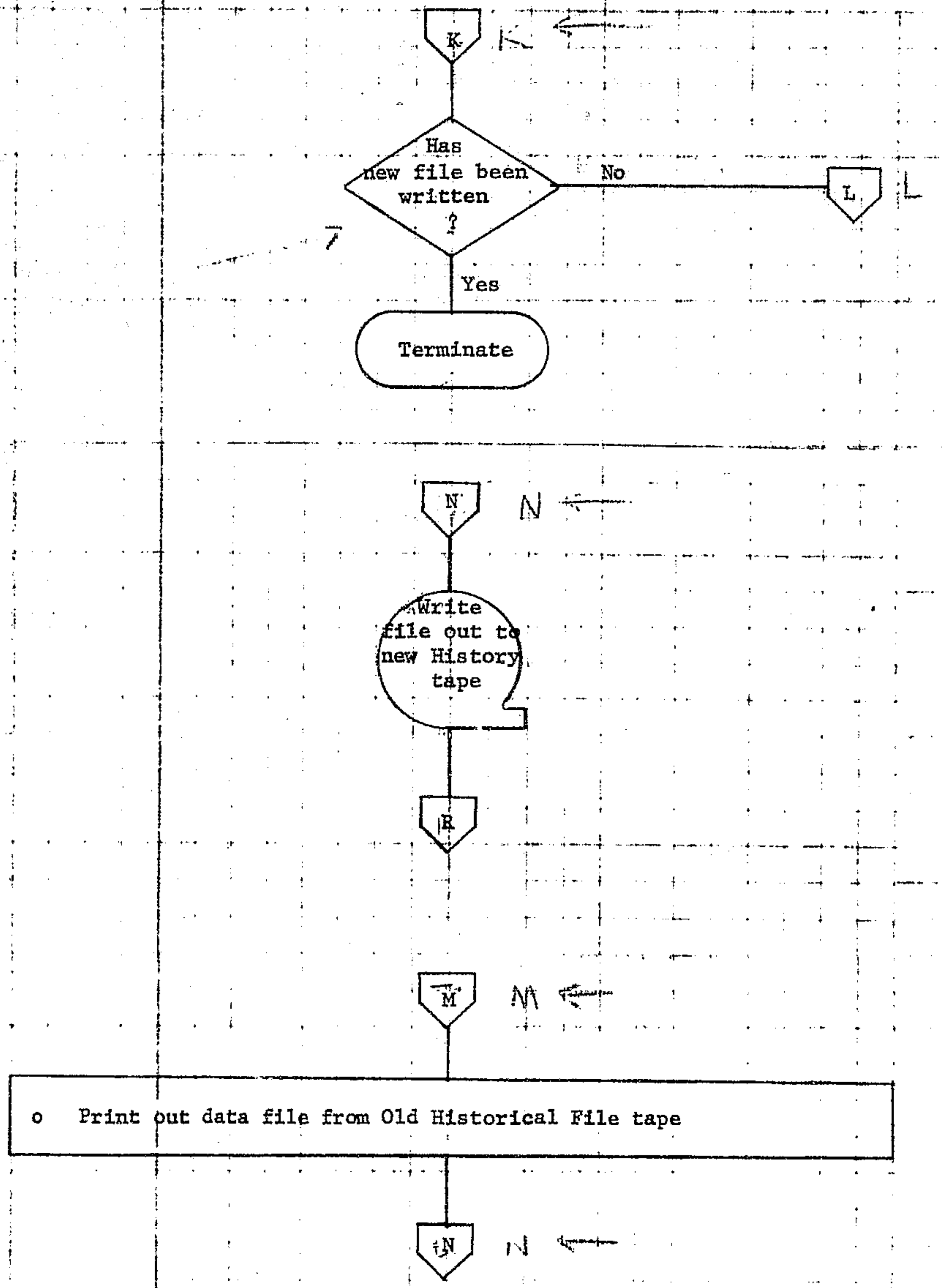
I


I
H
Process Bias Voltage Data ?
No
D
Yes
BIAPRO
o Initializes arrays for A1, A2, A3, A5 and A6 parameter data sample count
o Enumerates the data sample PCM count values within seven specified value ranges
o Rejects sync pulses from count
o Generates Histogram of the sample data
o Stores data for Historical file building
D
All Current Period Processed ?
No
A
Yes
All Periods Processed ?
Yes
J
No
o Update overall period start and stop times
A


C
ANPRO
o Determines if anomalies of a particular parameter have occurred on previous samples
o If previous anomalies have occurred, a counter is initiated or incremented, and a running total of anomalies, by parameter, is kept
o After a determined number of scans, or if the data interval terminates, the anomaly total for a particular parameter is computed
o If no anomaly occurred on the previous scan, the parameter PCM count value is converted to temperature or voltage
o Parameter anomaly data is printed out
o Parameter anomaly flags and counters are cleared.
B


L
Has new file been written ?
Yes
R
No
o Print new Historical data file
Write new data file out to new History tape
Q


J
QASUM
o Compiles a summary of the data derived during field, cal period, and wavelength cal period checks
o Prints out the summary of the data
R
Reads file from old history tape
Q
E.O.F. on old tape ?
Yes
K
No
Time interval for new history file ?
Yes
L
No
Should old and new file be printed ?
Yes
M
No
Delete old file ?
Yes
R
No
N


K
Has new file been written ?
No
L
Yes
Terminate
N
Write file out to new History tape
R
M
o Print out data file from Old Historical File tape
N

2.4.2 Inputs/Outputs

A. Inputs -S191H

1. 9-Track Time Edited Tape - (Reference Earth Resources Data Format Control Book, Section 2.2.4) The following is a measurement number list specifying units and digital range for each measurement.

ID	MEASUREMENT	UNITS	DIGITAL RANGE
A001-RRO	Long Wavelength Radiance, detector neg.	PCM CNTS	0-1023
A002-RRO	Short Wavelength Radiance, #3 detector	PCM CNTS	0-1023
A003-RRO	Short Wavelength Radiance, #1 detector	PCM CNTS	0-1023
A004-RRO	Filter Position Monitor	PCM CNTS	0-1023
A005-RRO	Short Wavelength Radiance, #2 detector	PCM CNTS	0-1023
A006-RRO	Long Wavelength Radiance, detector Pos.	PCM CNTS	0-1023
A007-RRO	Data Pallet Temp	PCM CNTS	0-1023
A008-RRO	Spectrometer Pallet Temp	PCM CNTS	0-1023
A009-RRO	Mirror Temperature	PCM CNTS	0-1023
A013-RRO	Heated Calibration temperature source	PCM CNTS	0-1023
A014-RRO	Ambient Calibration temp. source	PCM CNTS	0-1023
A015-RRO	Integrated Sphere Temp	PCM CNTS	0-1023
D005-RRO	Radiance Calibration wheel position	PCM CNTS	0-1023
D006-RRO	Field of View flag measurement	PCM CNTS	0-1023
D007-RRO	Zero Volts	PCM CNTS	0-1023
A016-RRO	Power Supply diagnostic	PCM CNTS	0-1023
A017-RRO	Thermal reference source temp.	PCM CNTS	0-1023
A018-RRO	Package temperature	PCM CNTS	0-1023
A019-RRO	Dichroic temperature	PCM CNTS	0-1023
A020-RRO	Long Wavelength detector temp	PCM CNTS	0-1023
A023-RRO	Calibration lamp current	PCM CNTS	0-1023

2. Control Input - Control input parameters are described in the lead card setup, Table 2.3.1-1.

3. Historical File Tape - A 9-track tape containing records of previously run QA191H Historical File in chronological order.

2.4.2 Inputs/Outputs (CONT.)

B. Outputs

Tabulation

1. QA191H ANOMALIES LIST

All parameters whose PCM count value exceeds the high or low tolerance limit will be printed out, along with the frame time, record number, and value of the data sample. During the calibration period, non-valid sync pulses and A4 parameter values that exceed the straight line value, plus or minus a tolerance, will also be printed out, with their associated frame time, record number, and data values.

2. QA191H TEST SUMMARY

A grand total of all anomalies detected will be listed, by parameter.

3. HISTORICAL FILE TAB

A listing of the values derived during calibration, wavelength calibration, and a histogram of the Bias Voltage PCM values is output. If no calibration or wavelength calibration period is on the processed tape, it will be so noted on the Historical File Tab. Upon request, all previously derived Historical Files on the Old Historical File Tape will also be tabbed out, as the new Historical File Tape is being built.

COMPUTER TAPES

1. A new Historical File Tape is generated at the end of each QA191H run. The new tape will contain all the files on the old Historical File Tape, as well as the new Historical File. The new file is inserted in the correct chronological order, based on the date of data generation.

2.4.3 Linkages (External)

Reference Earth Resources Processing Subsystem Support Software Documentation for linkage details. Document numbers are indicated below.

A. CONINP - Control Input Processor (ERS-300-01)

1. Input

- Data cards containing control input information - user input

- Block COMMON containing formatting and processing information related to the functional reading of the input cards and bounding information to facilitate the error processing of data read. The COMMON blocks which transmit information to CONINP are named /RDCNTL/ and /RDARG/

2. Output

- Data arrays containing S191H control input parameters. The named COMMONS /TIMES/, /TITLES/, and /INPUT/ contain the stored parameters read from the input cards

- An array of error codes transmitted by COMMON block /ERROR/ (maximum 20)

B. DCOM2N - Decommutation Processor Step 2 (ERS-300-02)

1. Input

- Start and stop times for extracting input data from the disk or 9-track tapes

- Array of measurement ID's for QA191H

- Maximum number of frames to retrieve

- Block COMMON containing format and processing information related to the functional reading of data from the disk or 9-track tape into core

2. Output

- Actual number of frames retrieved

- Time of last frame returned

- Data array for each measurement ID specified

- Array of error codes (maximum 20)

C. ERRPRC - Error Processor (ERS-300-07)

1. Input

- Error code array (maximum size 20 words)

2. Output

- Error messages
- Executes proper option for termination of job or returns to calling program

D. DCRIPT - Descriptor File Decoder for Non-Imagery Data Tape (ERS-300-02)

1. Input

- Descriptor record retrieval request flag
- The format number
- Indicator of input source as disk or tape
- Starting record where data is to be read
- The last record in the disk file to be read if applicable
- Logical unit number for the input tape or disk
- Size of the measurement array
- Block data containing the ASCII representation for each measurement code to be extracted
- Measurement list identifier

2. Output

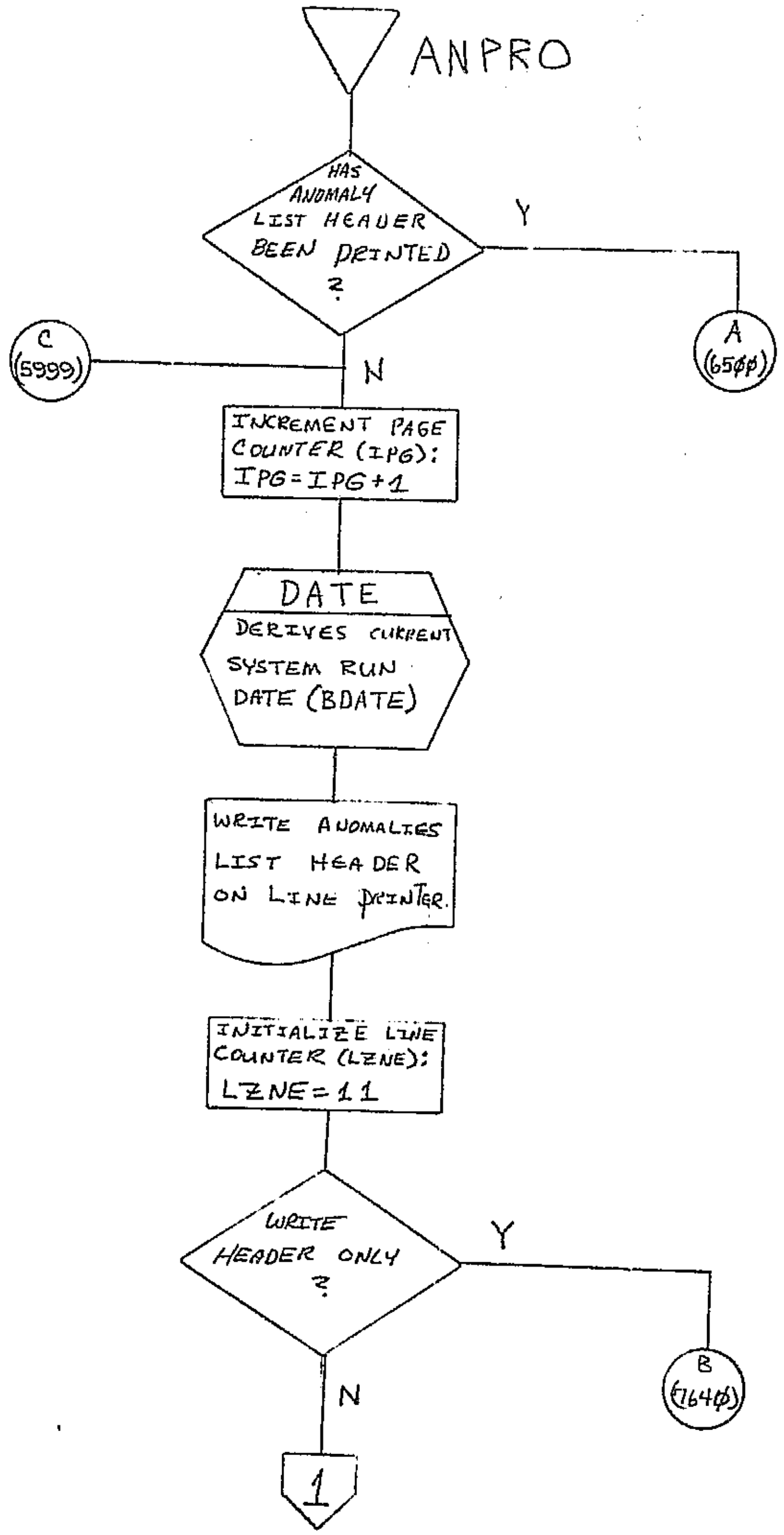
- An array containing descriptor file information for desired measurements
- The software routine status flag

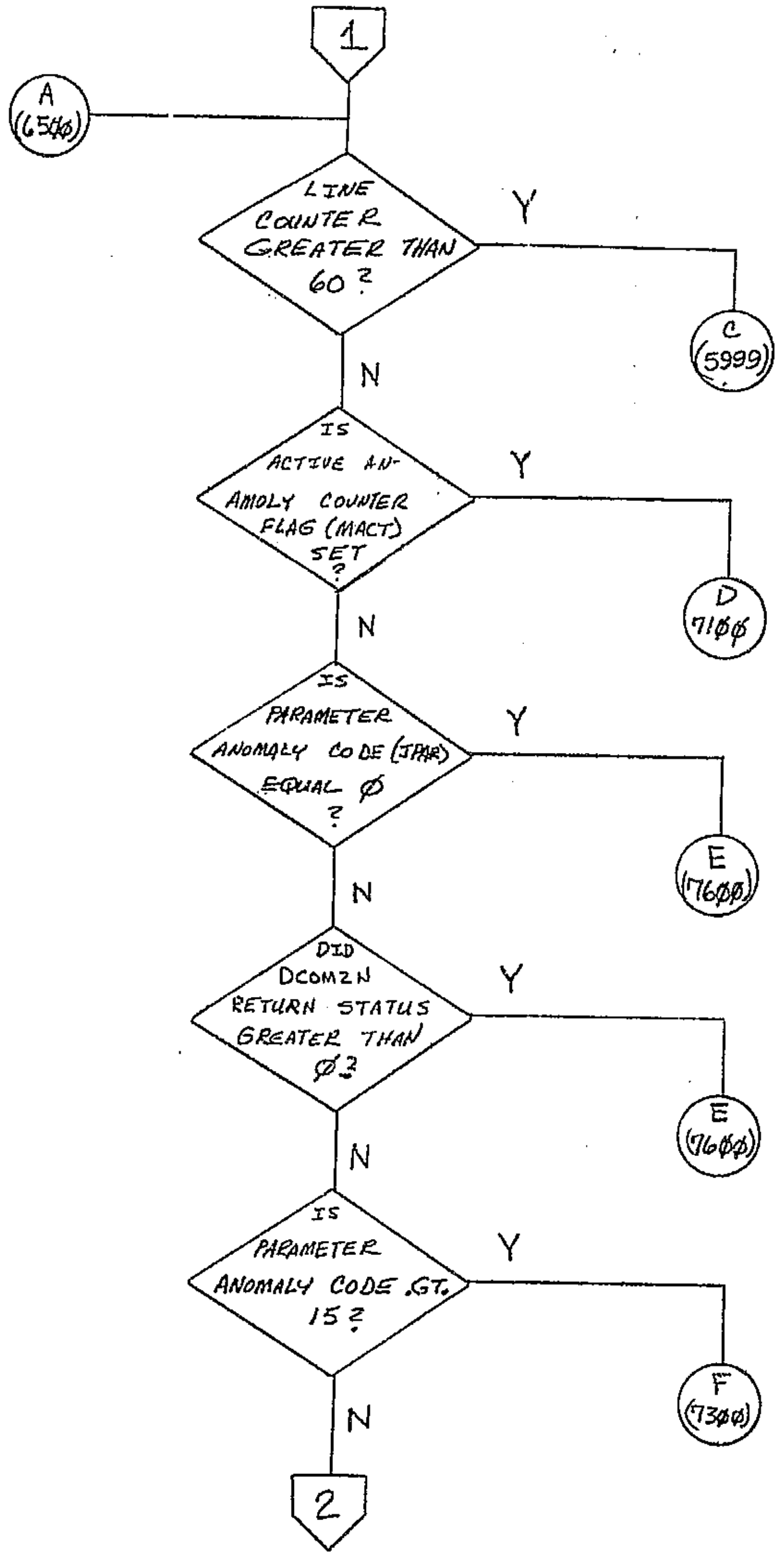
2.4.4 SPECIAL TECHNIQUES

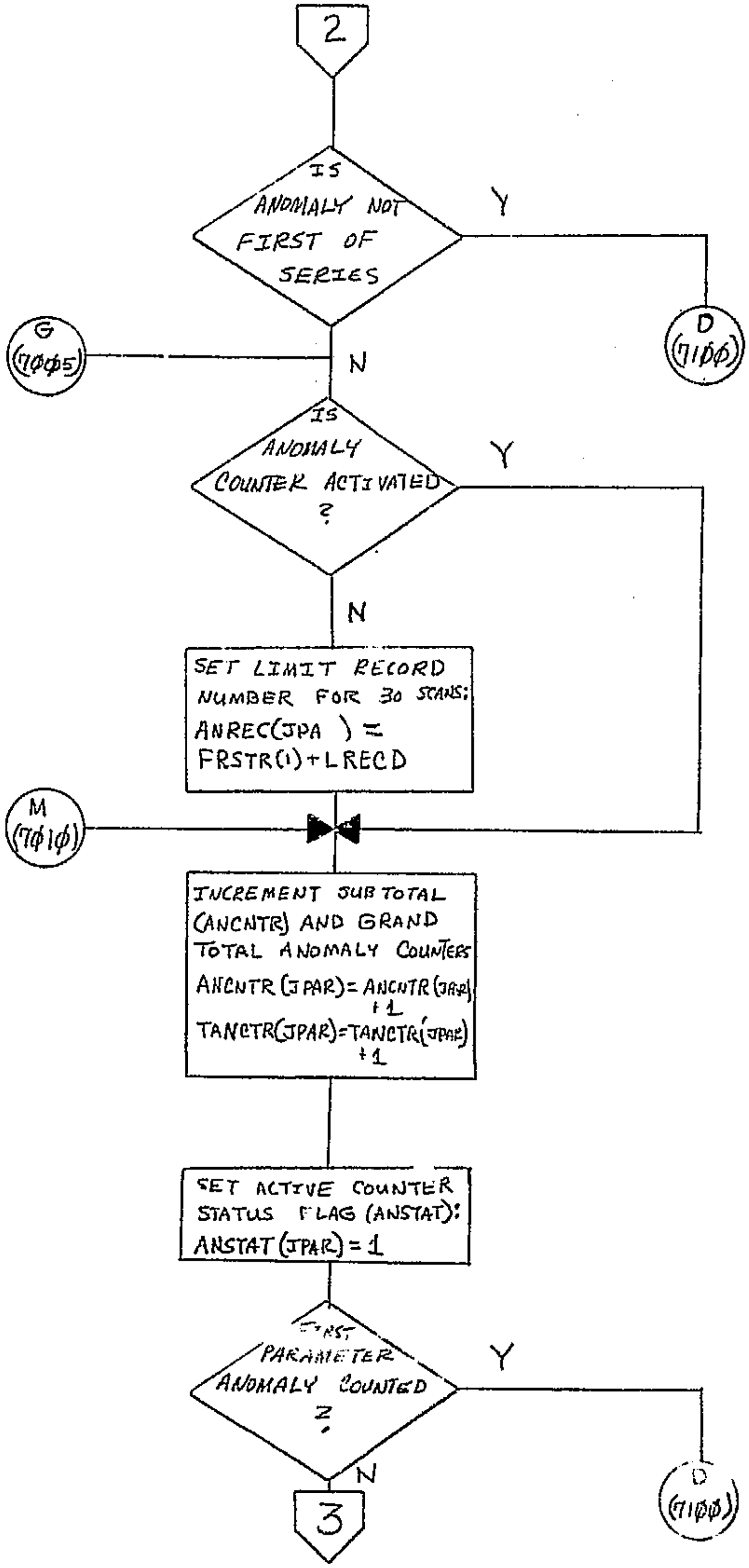
None required.

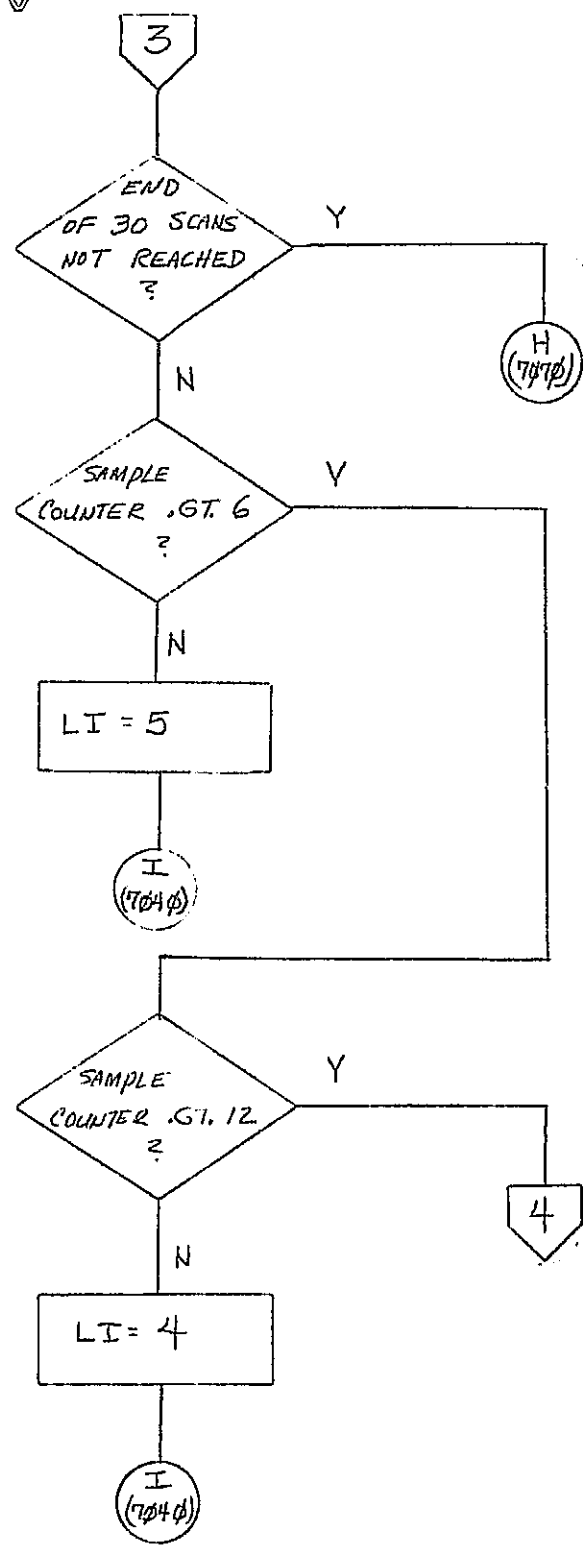
2.5 FLOW CHARTS

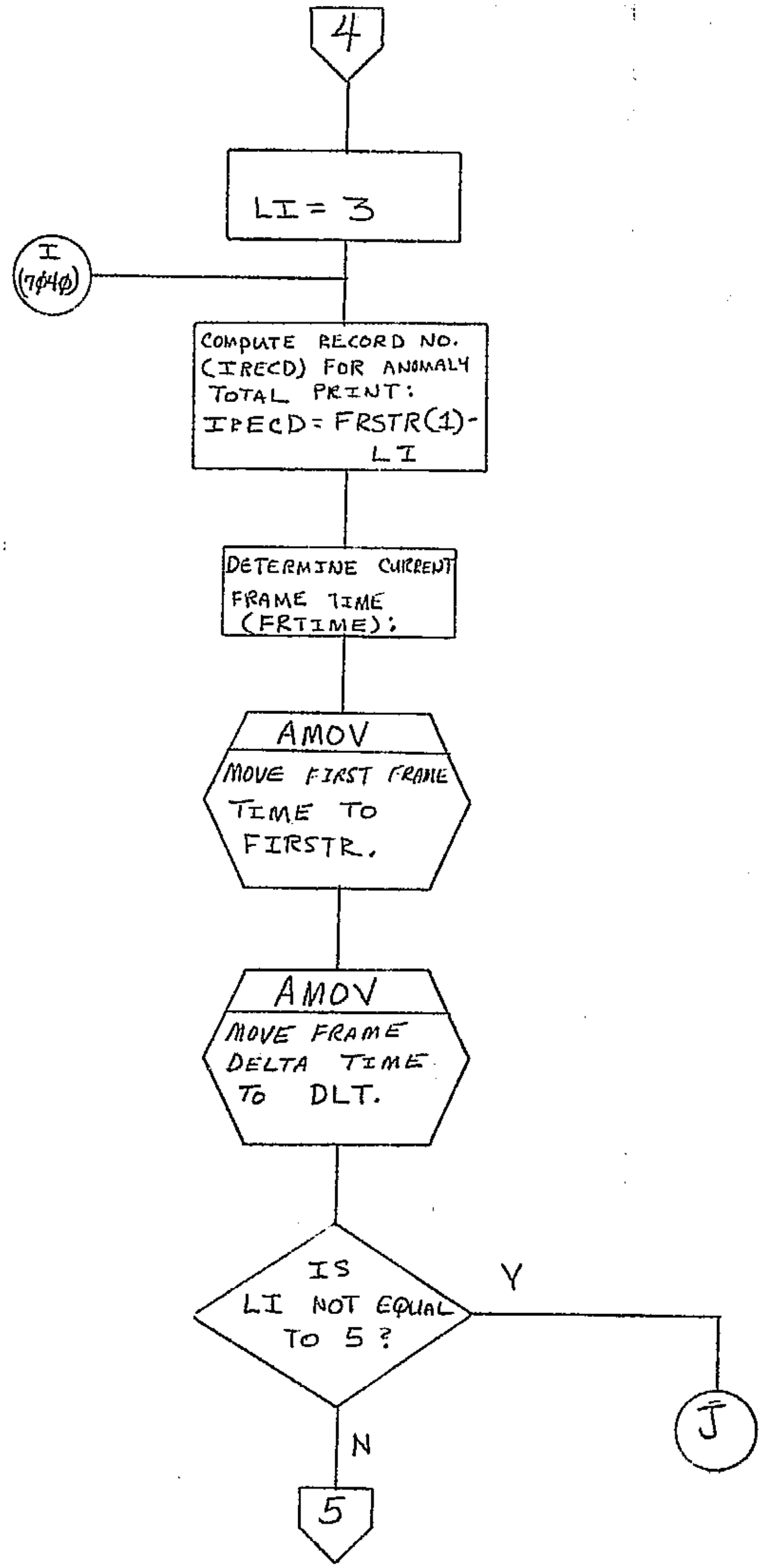
The detailed flow charts on the following pages reflect all program decision points, linkages with the operating system, and all significant computational and programming steps.

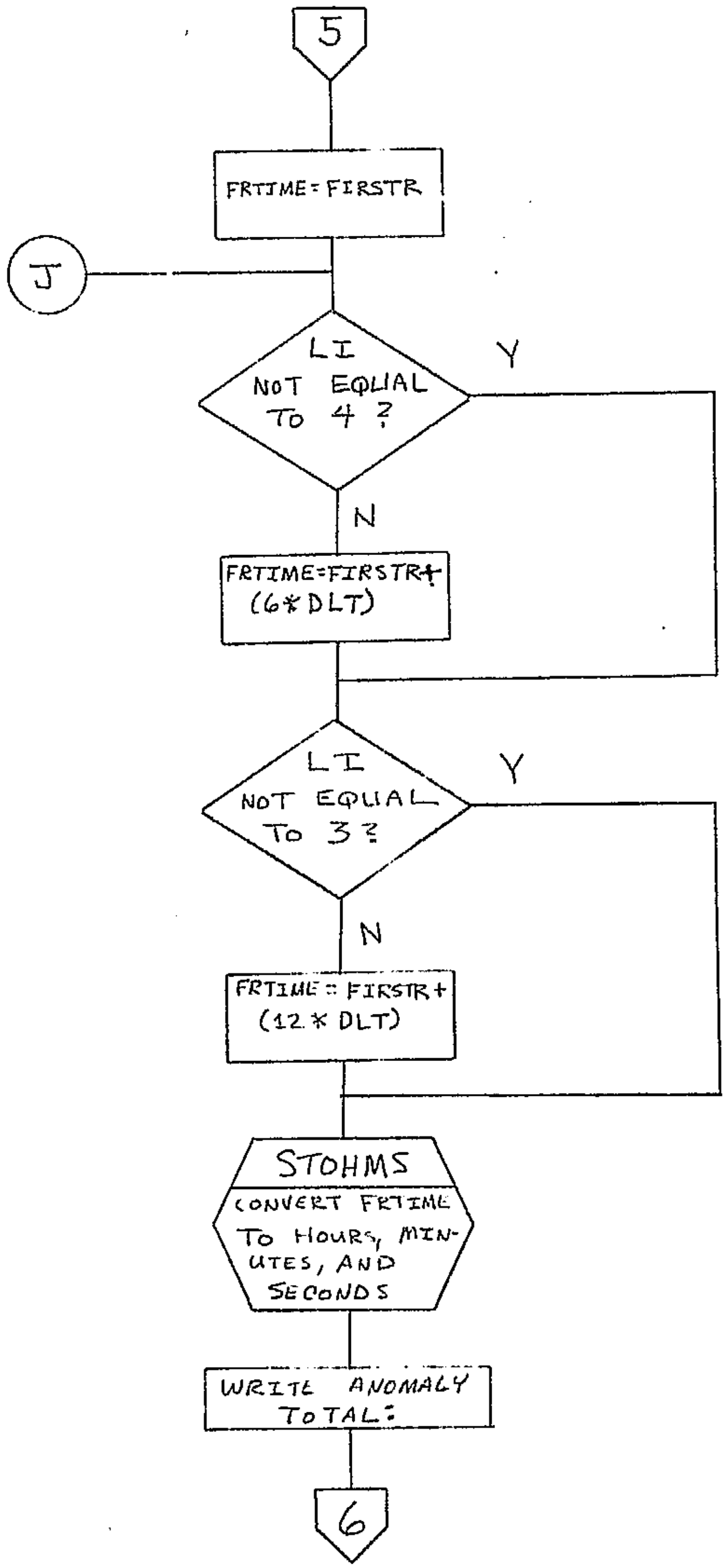

ANPRO
HAS ANOMALY LIST HEADER BEEN PRINTED ?
Y
A (65ØØ)
C (5999)
N
INCREMENT PAGE COUNTER (IPG): IPG=IPG+1
DATE
DERIVES CURRENT SYSTEM RUN DATE (BDATE)
WRITE ANOMALIES LIST HEADER ON LINE PRINTER.
INITIALIZE LINE COUNTER (LINE): LINE=11
WRITE HEADER ONLY ?
Y
B (764Ø)
N
1

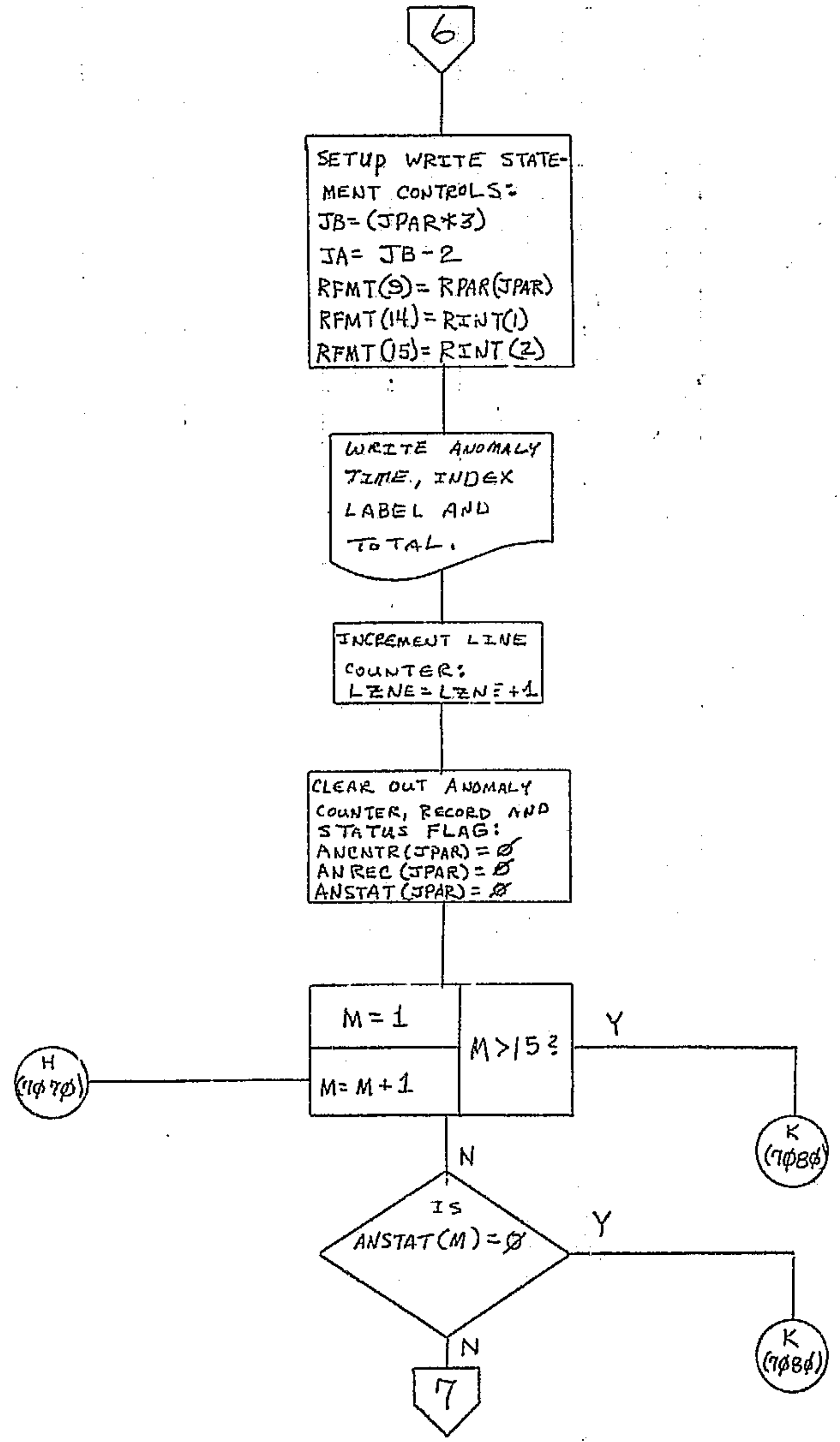

1
A (65ØØ)
LINE COUNTER GREATER THAN 60 ?
Y
C (5999)
N
IS ACTIVE ANAMOLY COUNTER FLAG (MACT) SET ?
Y
D 71ØØ
N
IS PARAMETER ANOMALY CODE (JPAR) EQUAL Ø ?
Y
E (76ØØ)
N
DID DCOMZN RETURN STATUS GREATER THAN Ø ?
Y
E (76ØØ)
N
IS PARAMETER ANOMALY CODE .GT. 15 ?
Y
F (73ØØ)
N
2

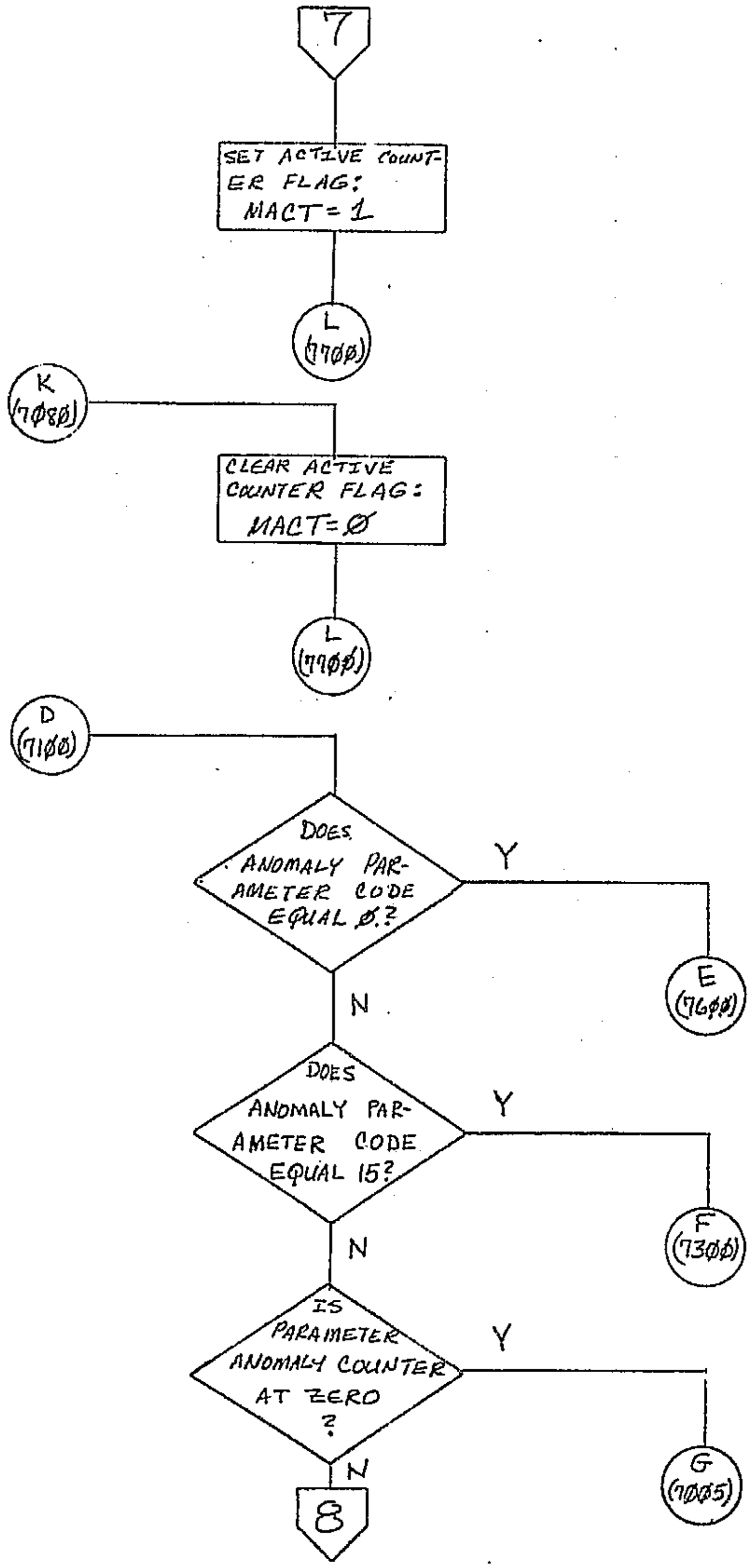

2
IS ANOMALY NOT FIRST OF SERIES
Y
D (71ϕϕ)
G (7ϕϕ5)
N
IS ANOMALY COUNTER ACTIVATED ?
Y
N
SET LIMIT RECORD NUMBER FOR 30 SCANS:
ANREC(JPA) = FRSTR(1)+LRECD
M (7ϕ1ϕ)
INCREMENT SUB TOTAL (ANCNTR) AND GRAND TOTAL ANOMALY COUNTERS
ANCNTR (JPAR) = ANCNTR (JPAR) +1
TANCTR(JPAR)=TANCTR(JPAR) +1
SET ACTIVE COUNTER STATUS FLAG (ANSTAT):
ANSTAT (JPAR) = 1
FIRST PARAMETER ANOMALY COUNTED ?
Y
N
3
D (71ϕϕ)

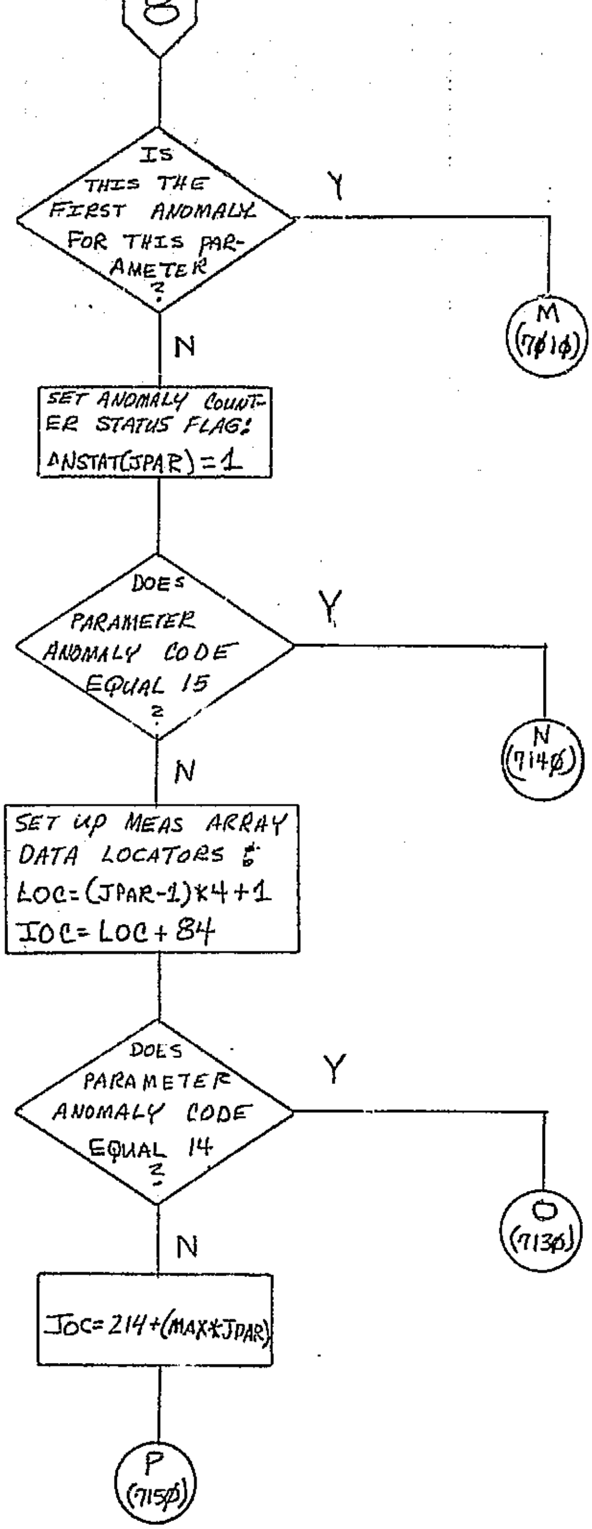

3
END OF 30 SCANS NOT REACHED ?
Y
H (7070)
N
SAMPLE COUNTER .GT. 6 ?
Y
N
LI = 5
I (7040)
SAMPLE COUNTER .GT. 12 ?
Y
4
N
LI = 4
I (7040)

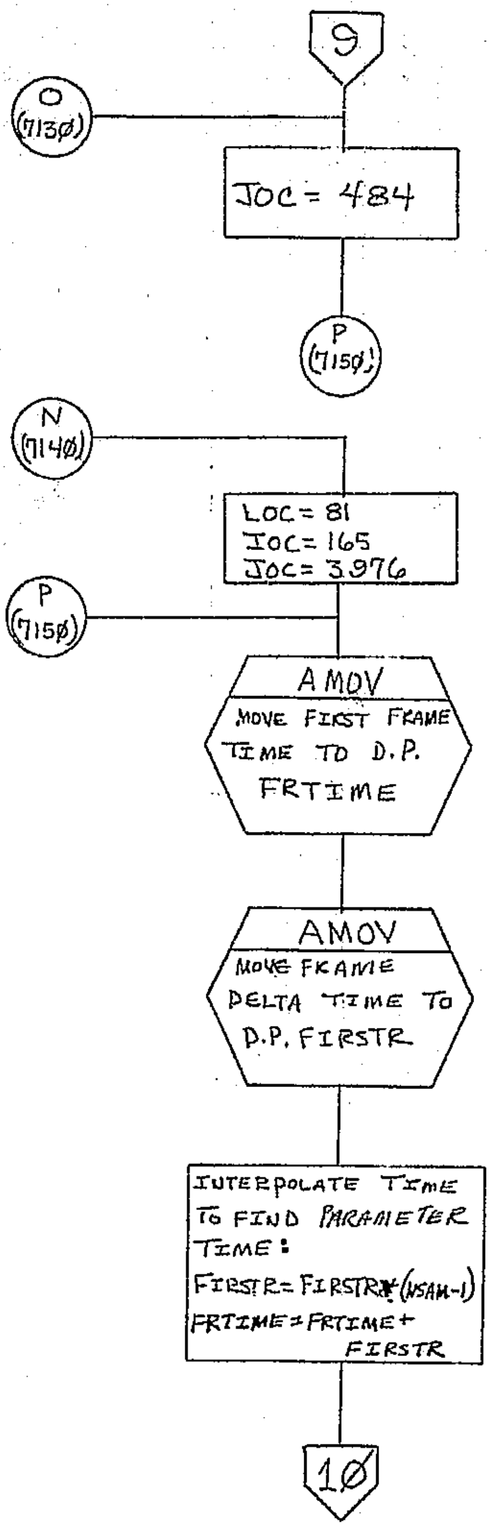

4
LI = 3
I (7040)
COMPUTE RECORD NO. (IRECD) FOR ANOMALY TOTAL PRINT:
IRECD = FRSTR(1)-LI
DETERMINE CURRENT FRAME TIME (FRTIME):
AMOV
MOVE FIRST FRAME TIME TO FIRSTR.
AMOV
MOVE FRAME DELTA TIME TO DLT.
IS LI NOT EQUAL TO 5 ?
Y
J
N
5

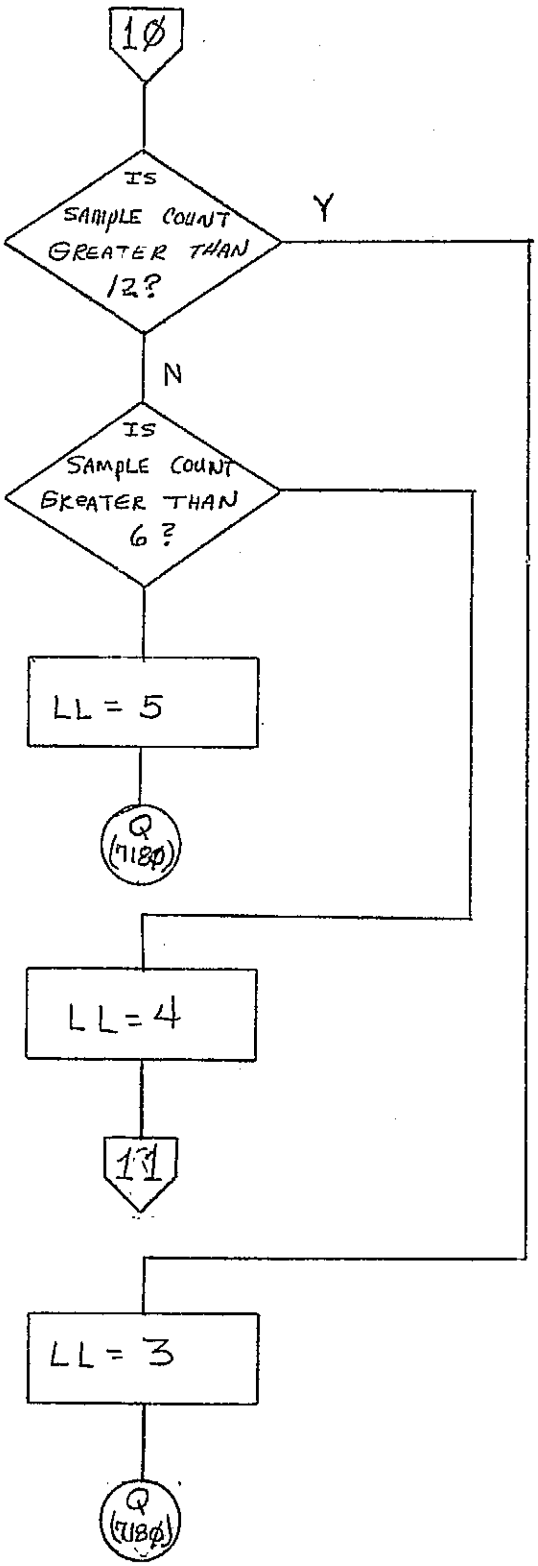

5
FRTIME = FIRSTR
J
LI NOT EQUAL TO 4?
Y
N
FRTIME=FIRSTR+ (6*DLT)
LI NOT EQUAL TO 3?
Y
N
FRTIME = FIRSTR+ (12 * DLT)
STOHMS
CONVERT FRTIME TO HOURS, MIN-UTES, AND SECONDS
WRITE ANOMALY TOTAL:
6

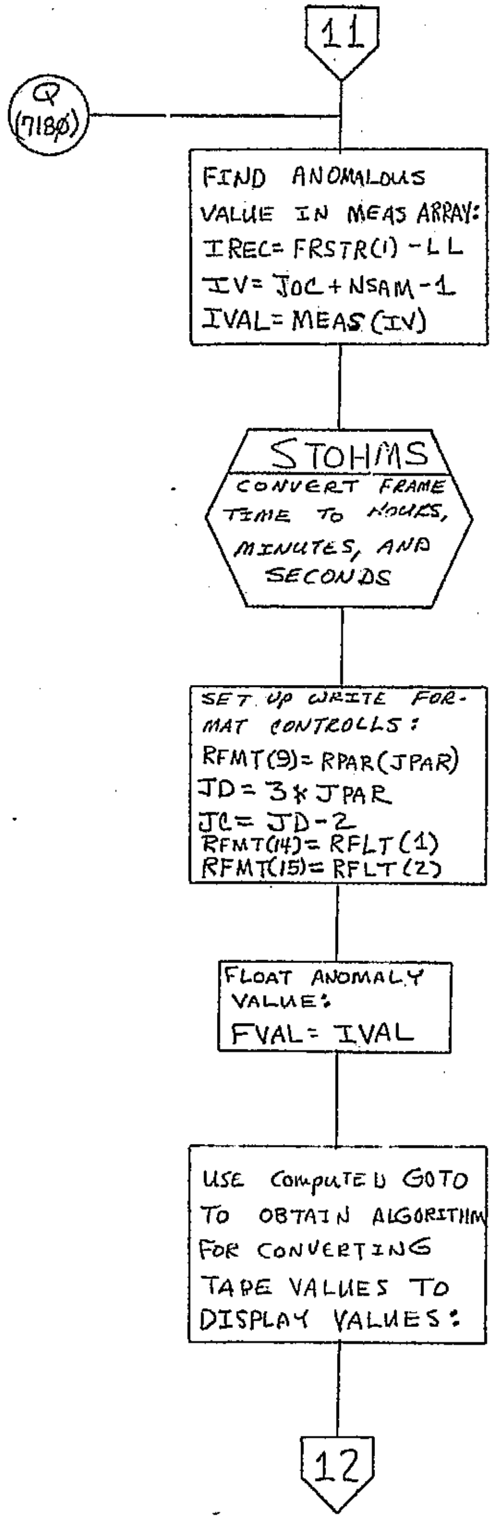

6
SETUP WRITE STATEMENT CONTROLS:
JB=(JPAR*3)
JA= JB-2
RFMT(9)= RPAR(JPAR)
RFMT(14)= RINT(1)
RFMT(15)= RINT(2)
WRITE ANOMALY TIME, INDEX LABEL AND TOTAL.
INCREMENT LINE COUNTER:
LINE=LINE+1
CLEAR OUT ANOMALY COUNTER, RECORD AND STATUS FLAG:
ANCNTR(JPAR)=Ø
ANREC(JPAR)=Ø
ANSTAT(JPAR)=Ø
M=1
M=M+1
M>15?
Y
N
H
(1Ø7Ø)
K
(7Ø8Ø)
IS ANSTAT(M)=Ø
Y
N
K
(7Ø8Ø)
7

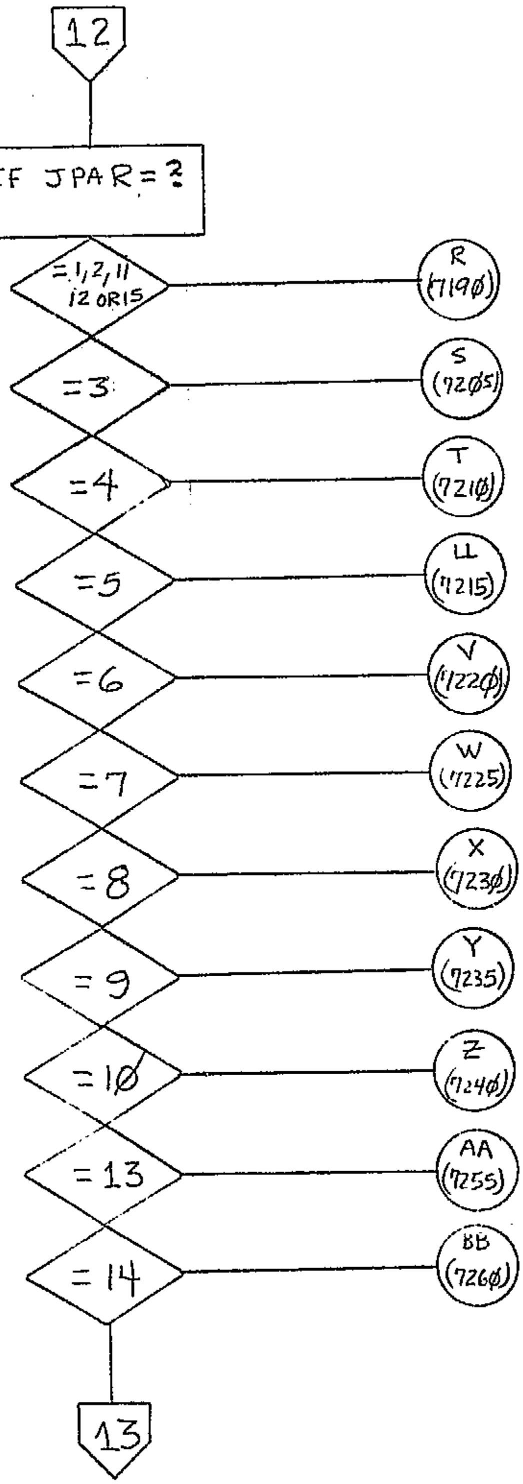

7
SET ACTIVE COUNTER FLAG:
MACT = 1
L (7700)
K (7080)
CLEAR ACTIVE COUNTER FLAG:
MACT = Ø
L (7700)
D (7100)
DOES ANOMALY PARAMETER CODE EQUAL Ø?
Y
E (7600)
N
DOES ANOMALY PARAMETER CODE EQUAL 15?
Y
F (7300)
N
IS PARAMETER ANOMALY COUNTER AT ZERO ?
Y
G (7005)
N
8

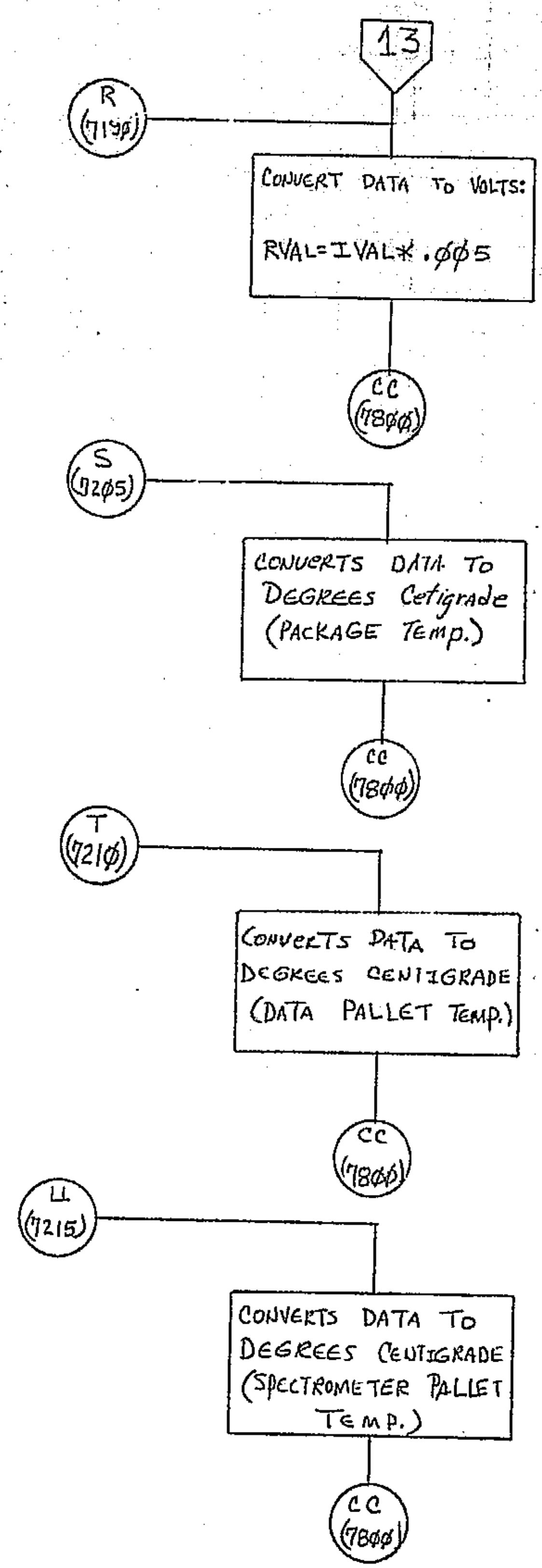

8
IS THIS THE FIRST ANOMALY FOR THIS PARAMETER ?
Y
M (7010)
N
SET ANOMALY COUNTER STATUS FLAG:
ANSTAT(JPAR) = 1
DOES PARAMETER ANOMALY CODE EQUAL 15 ?
Y
N (7140)
N
SET UP MEAS ARRAY DATA LOCATORS &
LOC = (JPAR-1)*4+1
IOC = LOC + 84
DOES PARAMETER ANOMALY CODE EQUAL 14 ?
Y
O (7130)
N
JOC = 214+(MAX*JPAR)
P (7150)

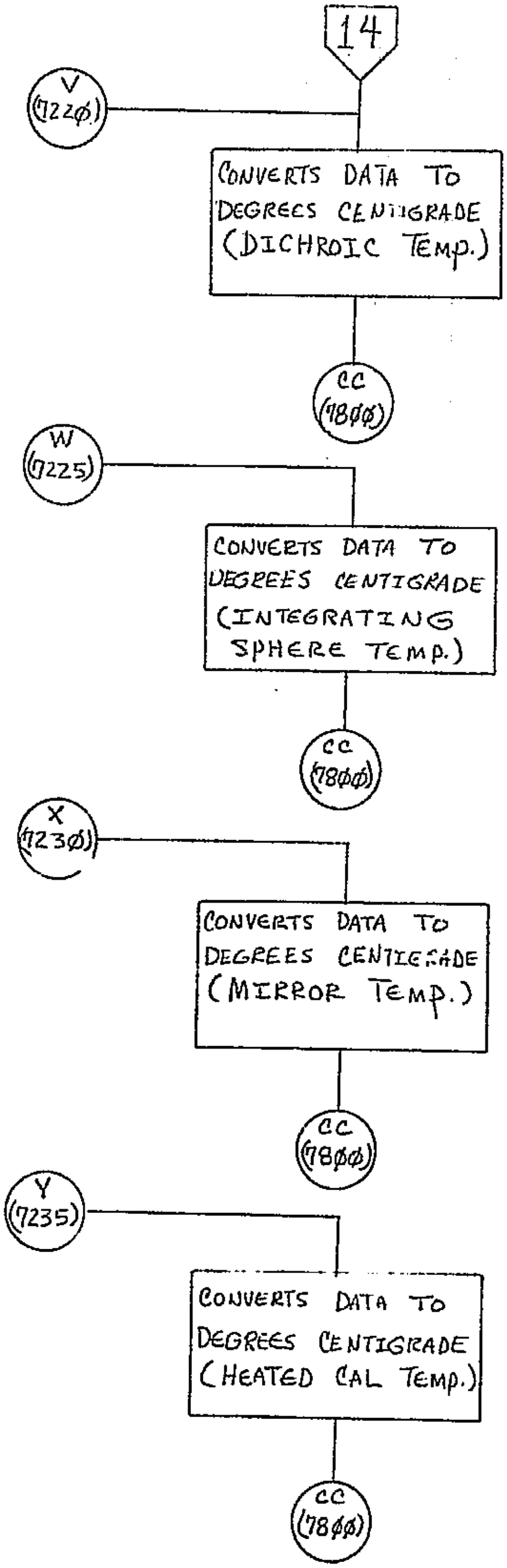

9
O (713Ø)
JOC = 484
P (715Ø)
N (714Ø)
LOC = 81
IOC = 165
JOC = 3976
P (715Ø)
AMOV
MOVE FIRST FRAME TIME TO D.P. FRTIME
AMOV
MOVE FRAME DELTA TIME TO D.P. FIRSTR
INTERPOLATE TIME TO FIND PARAMETER TIME:
FIRSTR = FIRSTR*(NSAM-1)
FRTIME = FRTIME + FIRSTR
1Ø

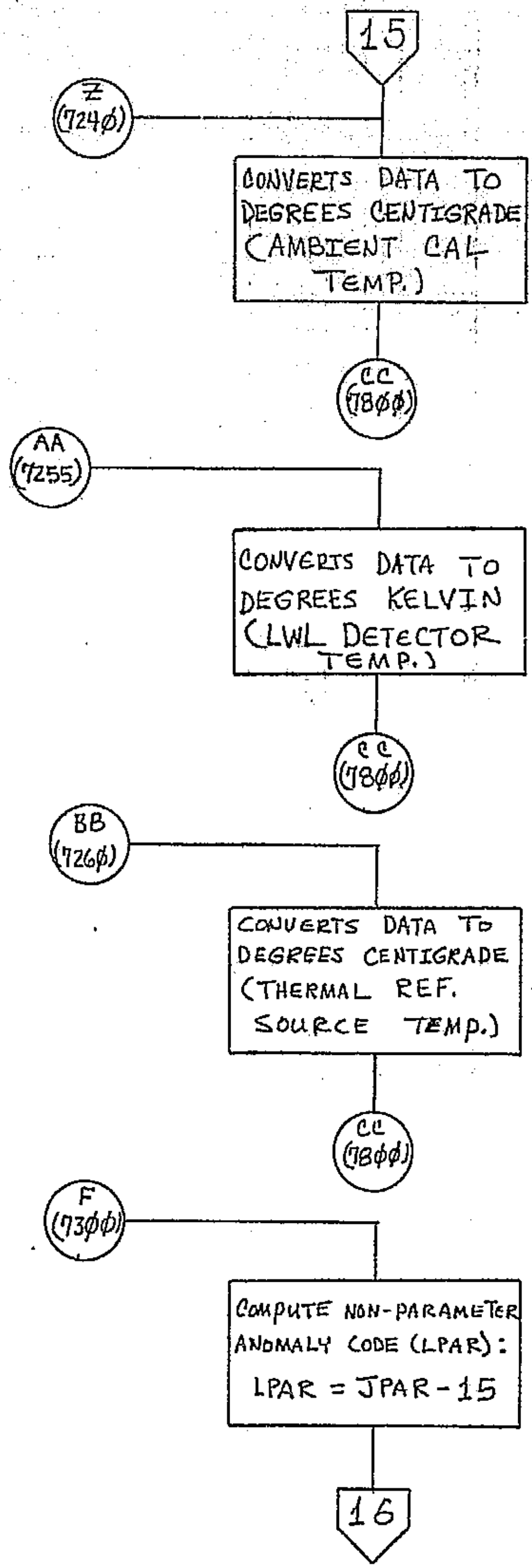

10
IS SAMPLE COUNT GREATER THAN 12?
Y
N
IS SAMPLE COUNT GREATER THAN 6?
LL = 5
Q (7180)
LL=4
11
LL = 3
Q (7180)

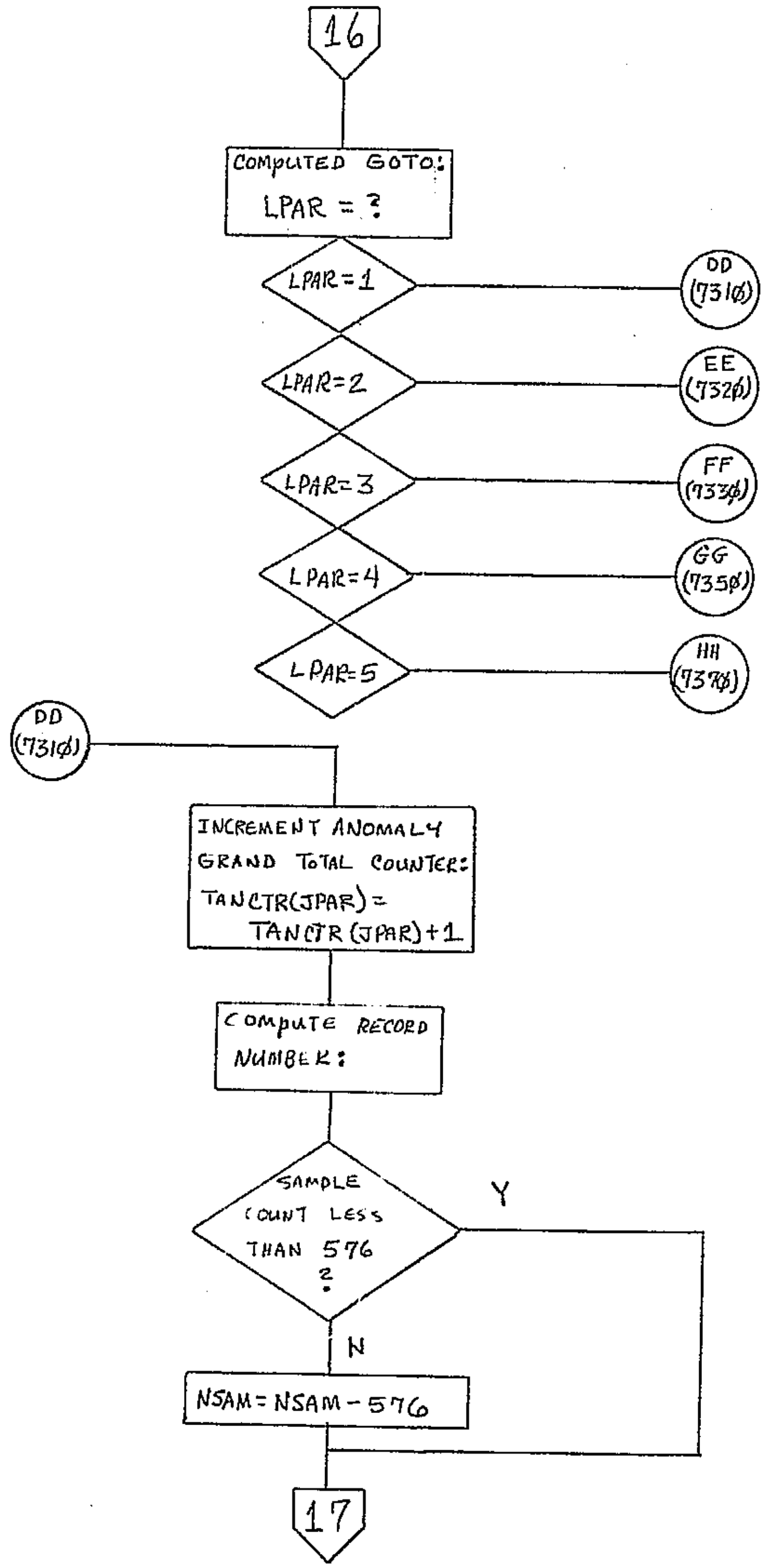

11
Q
(718Ø)
FIND ANOMALOUS VALUE IN MEAS ARRAY:
IREC= FRSTR(1) -LL
IV= JOC + NSAM -1
IVAL= MEAS (IV)
STOHMS
CONVERT FRAME TIME TO HOURS, MINUTES, AND SECONDS
SET UP WRITE FORMAT CONTROLLS:
RFMT(9)= RPAR(JPAR)
JD= 3 * JPAR
JC= JD-2
RFMT(14)= RFLT(1)
RFMT(15)= RFLT(2)
FLOAT ANOMALY VALUE:
FVAL= IVAL
USE COMPUTED GOTO TO OBTAIN ALGORITHM FOR CONVERTING TAPE VALUES TO DISPLAY VALUES:
12

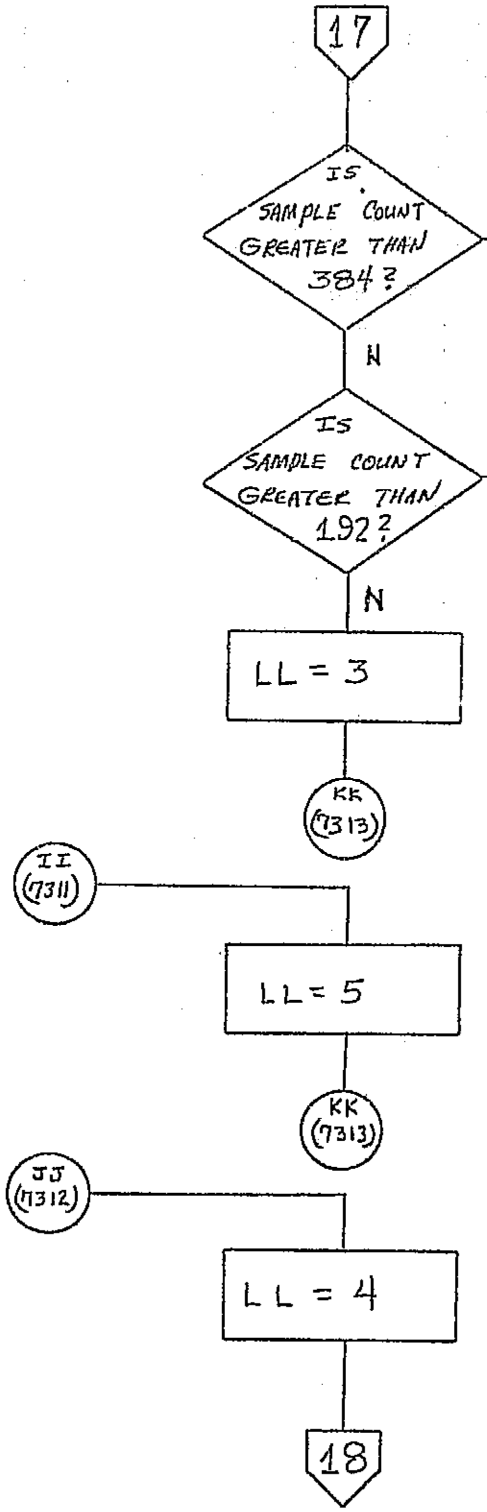

12
IF JPAR = ?
=1,2,11
12 OR 15
R
(719Ø)
=3
S
(72Ø5)
=4
T
(721Ø)
=5
U
(7215)
=6
V
(722Ø)
=7
W
(7225)
=8
X
(723Ø)
=9
Y
(7235)
=1Ø
Z
(724Ø)
=13
AA
(7255)
=14
BB
(726Ø)
13

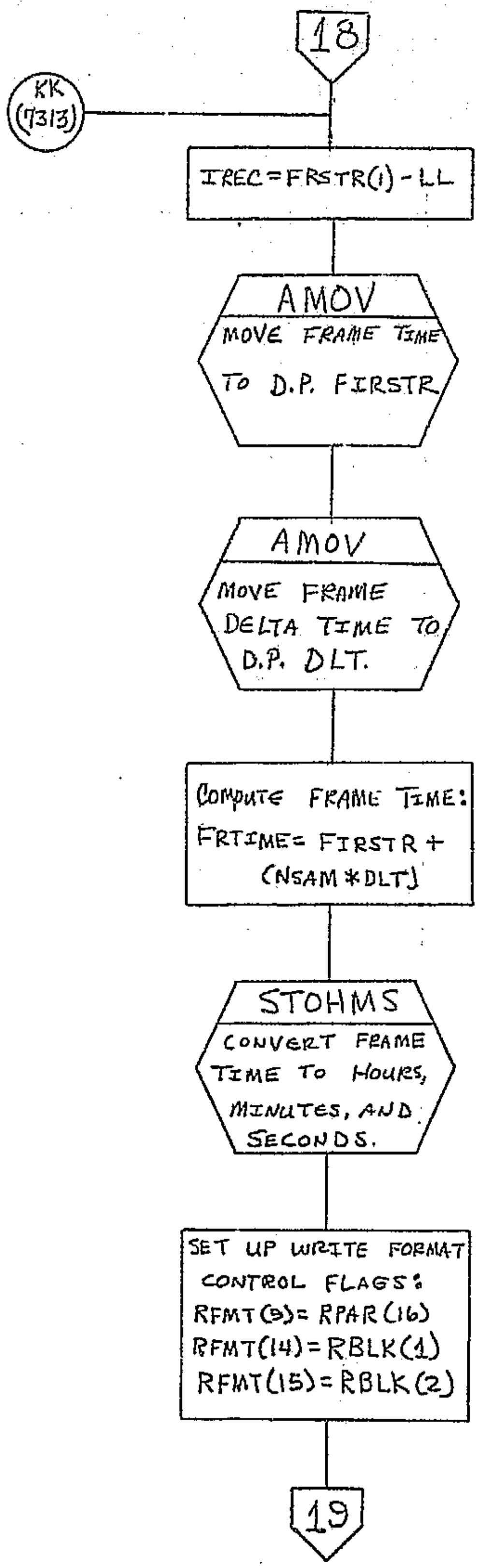

13
R
(719∅)
CONVERT DATA TO VOLTS:
RVAL=IVAL*.∅∅5
CC
(78∅∅)
S
(72∅5)
CONVERTS DATA TO DEGREES Cetigrade (PACKAGE Temp.)
CC
(78∅∅)
T
(721∅)
CONVERTS DATA TO DEGREES CENTIGRADE (DATA PALLET Temp.)
CC
(78∅∅)
U
(7215)
CONVERTS DATA TO DEGREES CENTIGRADE (SPECTROMETER PALLET TEMP.)
CC
(78∅∅)

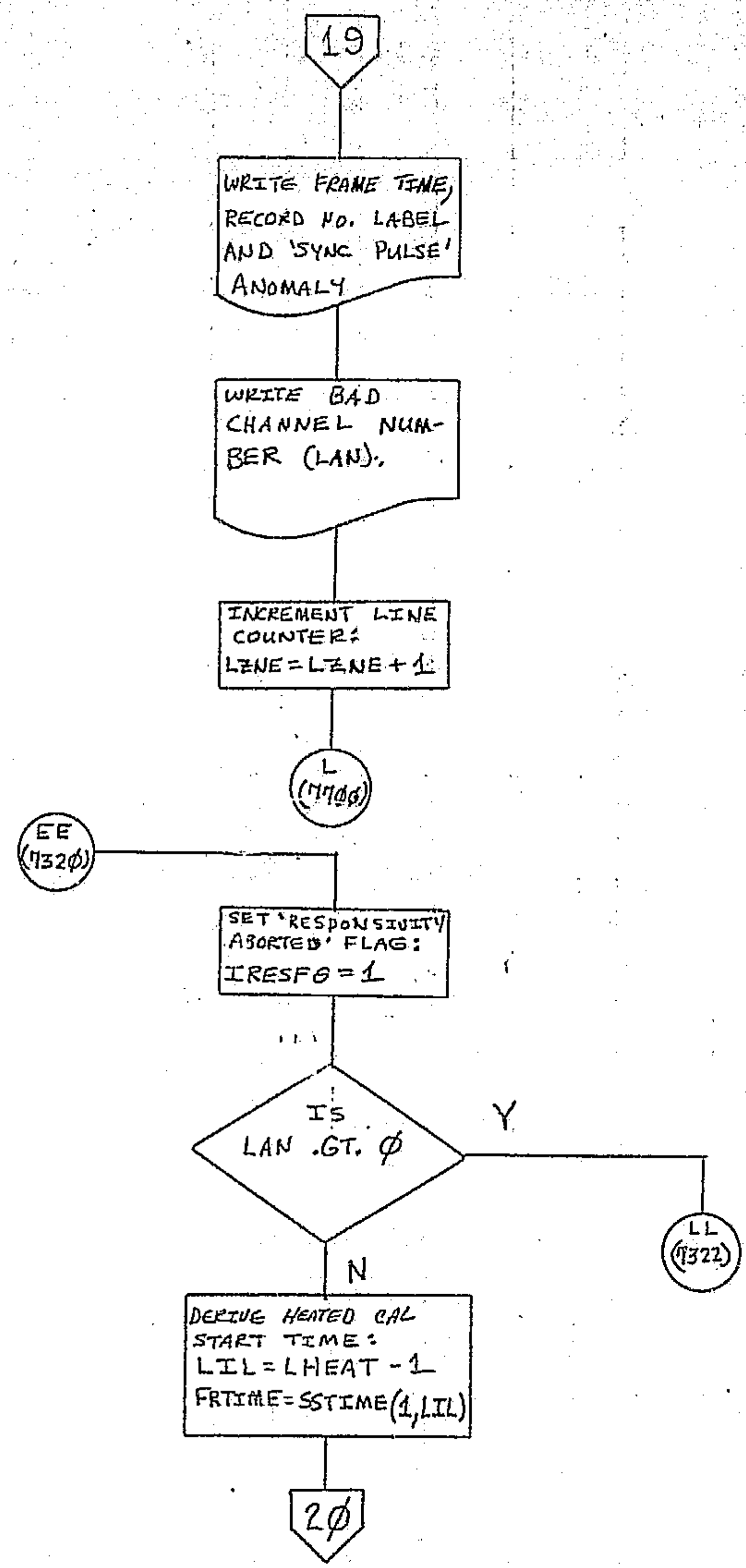

14
V
(722φ)
CONVERTS DATA TO
DEGREES CENTIGRADE
(DICHROIC TEMP.)
CC
(78φφ)
W
(7225)
CONVERTS DATA TO
DEGREES CENTIGRADE
(INTEGRATING
SPHERE TEMP.)
CC
(78φφ)
X
(723φ)
CONVERTS DATA TO
DEGREES CENTIGRADE
(MIRROR TEMP.)
CC
(78φφ)
Y
(7235)
CONVERTS DATA TO
DEGREES CENTIGRADE
(HEATED CAL TEMP.)
CC
(78φφ)

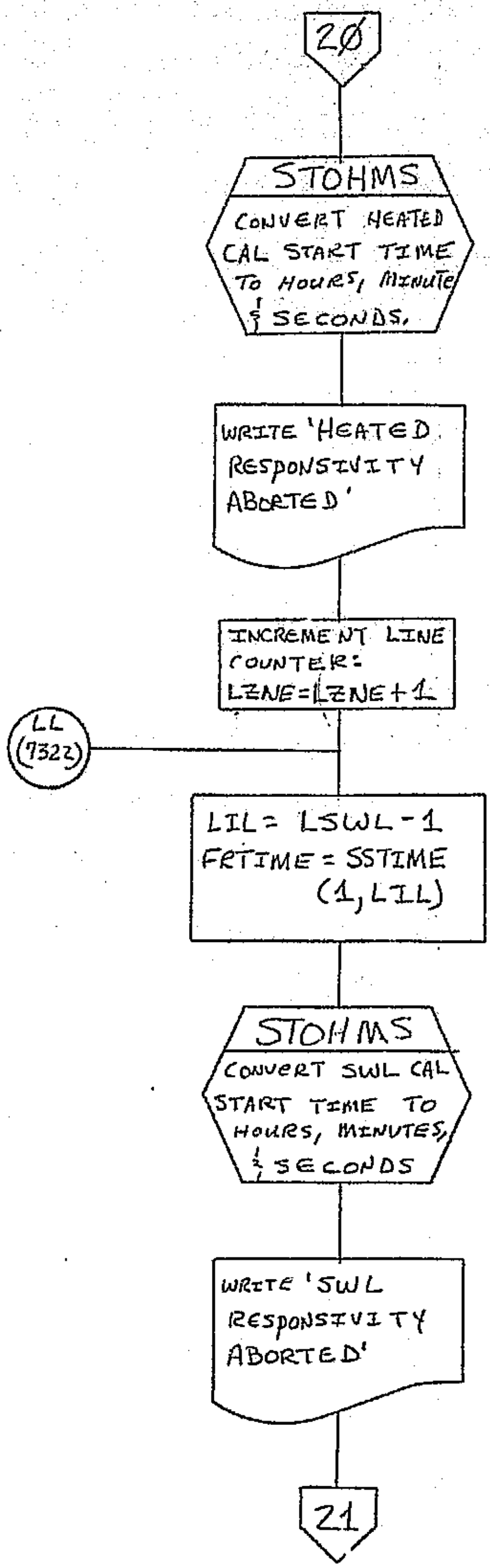

15
Z (724Ø)
CONVERTS DATA TO DEGREES CENTIGRADE (AMBIENT CAL TEMP.)
CC (78ØØ)
AA (7255)
CONVERTS DATA TO DEGREES KELVIN (LWL DETECTOR TEMP.)
CC (78ØØ)
BB (726Ø)
CONVERTS DATA TO DEGREES CENTIGRADE (THERMAL REF. SOURCE TEMP.)
CC (78ØØ)
F (73ØØ)
COMPUTE NON-PARAMETER ANOMALY CODE (LPAR):
LPAR = JPAR - 15
16

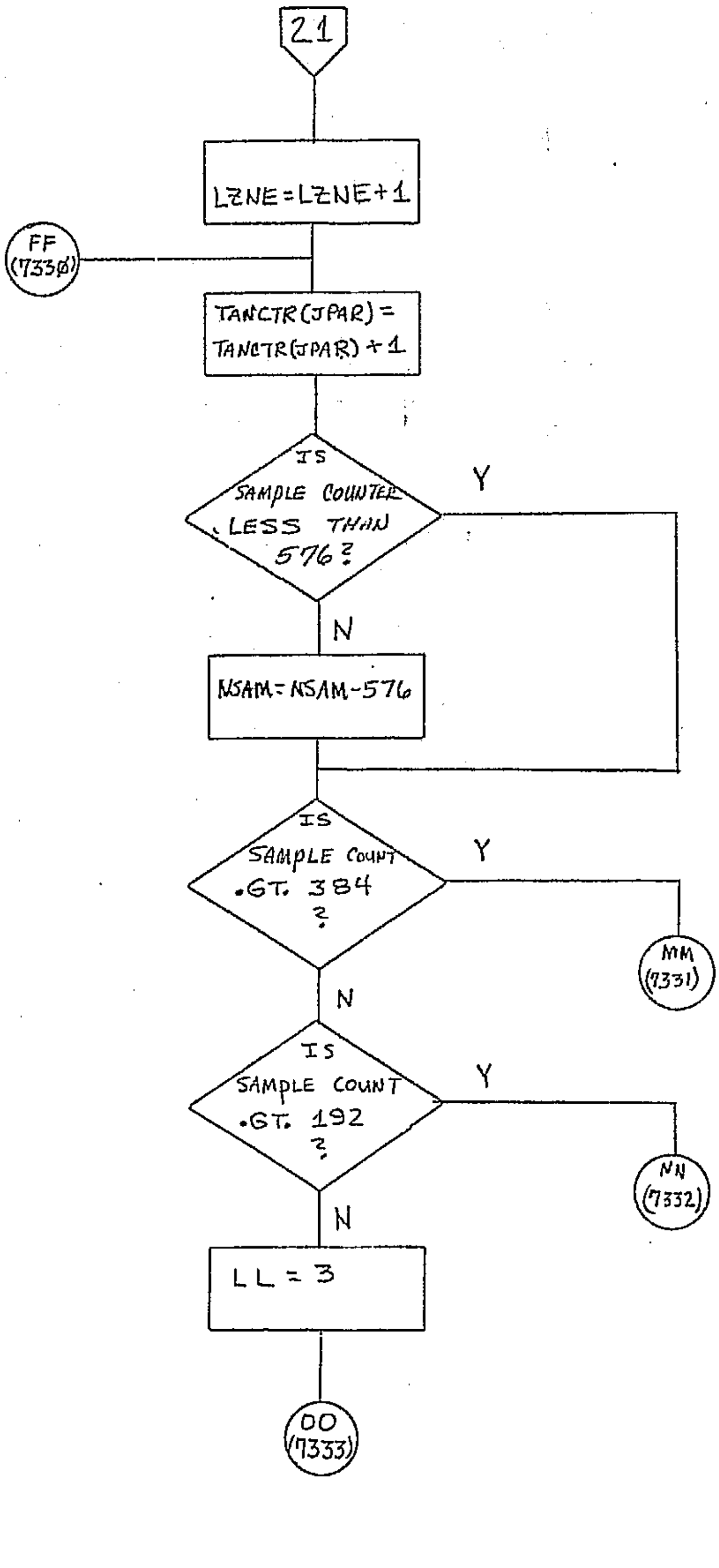

16
COMPUTED GOTO:
LPAR = ?
LPAR = 1
DD (731ø)
LPAR=2
EE (732ø)
LPAR=3
FF (733ø)
LPAR=4
GG (735ø)
LPAR=5
HH (737ø)
DD (731ø)
INCREMENT ANOMALY GRAND TOTAL COUNTER:
TANCTR(JPAR) = TANCTR (JPAR)+1
COMPUTE RECORD NUMBER:
SAMPLE COUNT LESS THAN 576 ?
Y
N
NSAM=NSAM-576
17

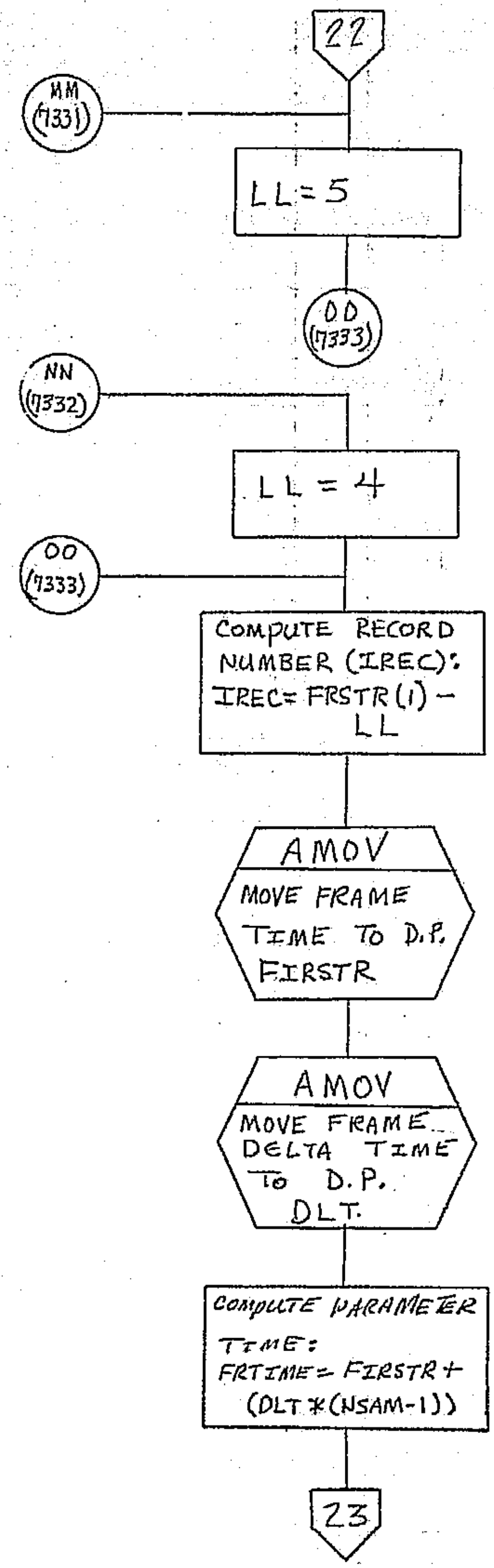

17
IS SAMPLE COUNT GREATER THAN 384?
Y
N
II (7311)
IS SAMPLE COUNT GREATER THAN 192?
Y
N
JJ (7312)
LL = 3
KK (7313)
II (7311)
LL = 5
KK (7313)
JJ (7312)
LL = 4
18

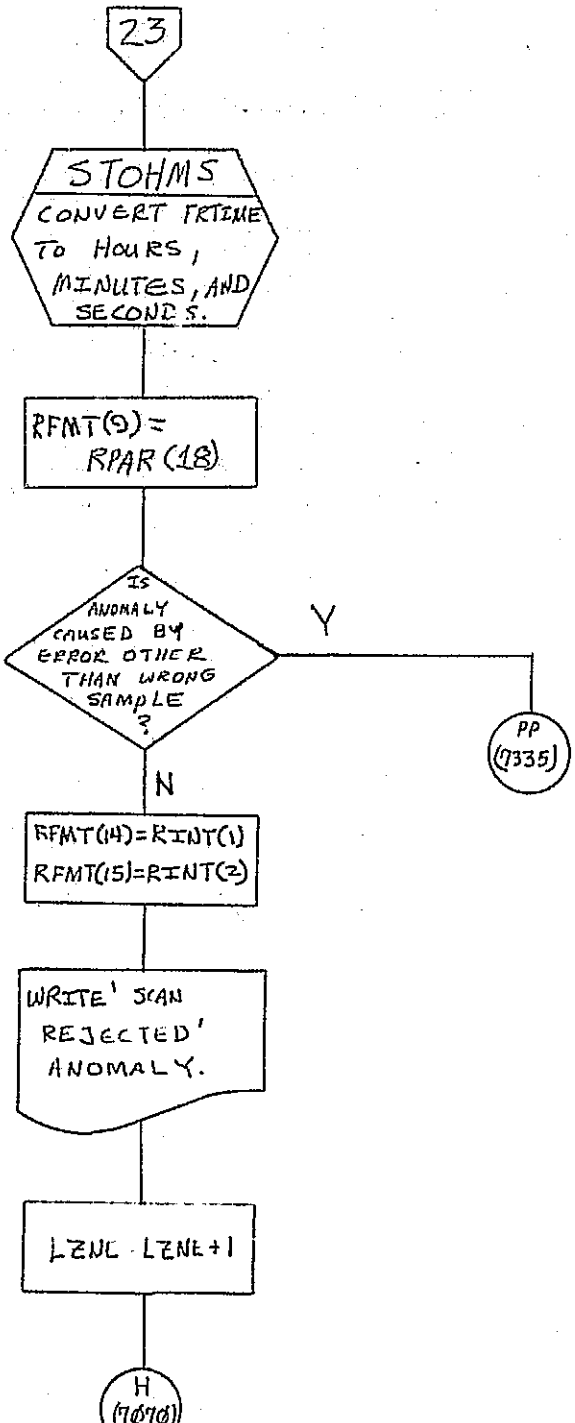

18
KK (7313)
IREC=FRSTR(1)-LL
AMOV
MOVE FRAME TIME TO D.P. FIRSTR
AMOV
MOVE FRAME DELTA TIME TO D.P. DLT.
COMPUTE FRAME TIME:
FRTIME= FIRSTR + (NSAM *DLT)
STOHMS
CONVERT FRAME TIME TO HOURS, MINUTES, AND SECONDS.
SET UP WRITE FORMAT CONTROL FLAGS:
RFMT(9)= RPAR(16)
RFMT(14)=RBLK(1)
RFMT(15)=RBLK(2)
19

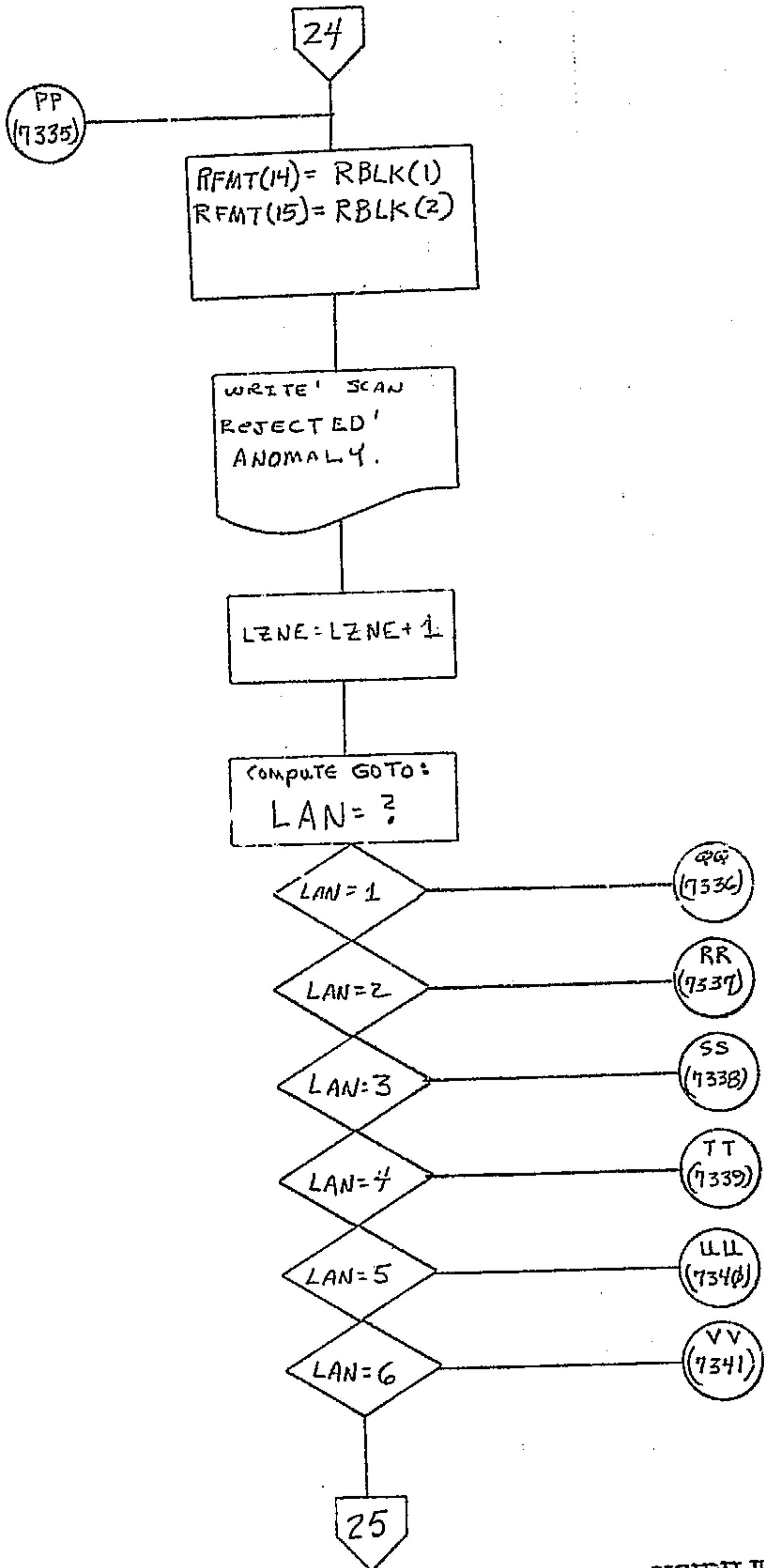

19
WRITE FRAME TIME, RECORD NO. LABEL AND 'SYNC PULSE' ANOMALY
WRITE BAD CHANNEL NUMBER (LAN).
INCREMENT LINE COUNTER: LINE = LINE + 1
L (770Ø)
EE (732Ø)
SET 'RESPONSIVITY ABORTED' FLAG: IRESFG = 1
IS LAN .GT. Ø
Y
LL (7322)
N
DERIVE HEATED CAL START TIME: LIL = LHEAT - 1 FRTIME = SSTIME(1,LIL)
2Ø

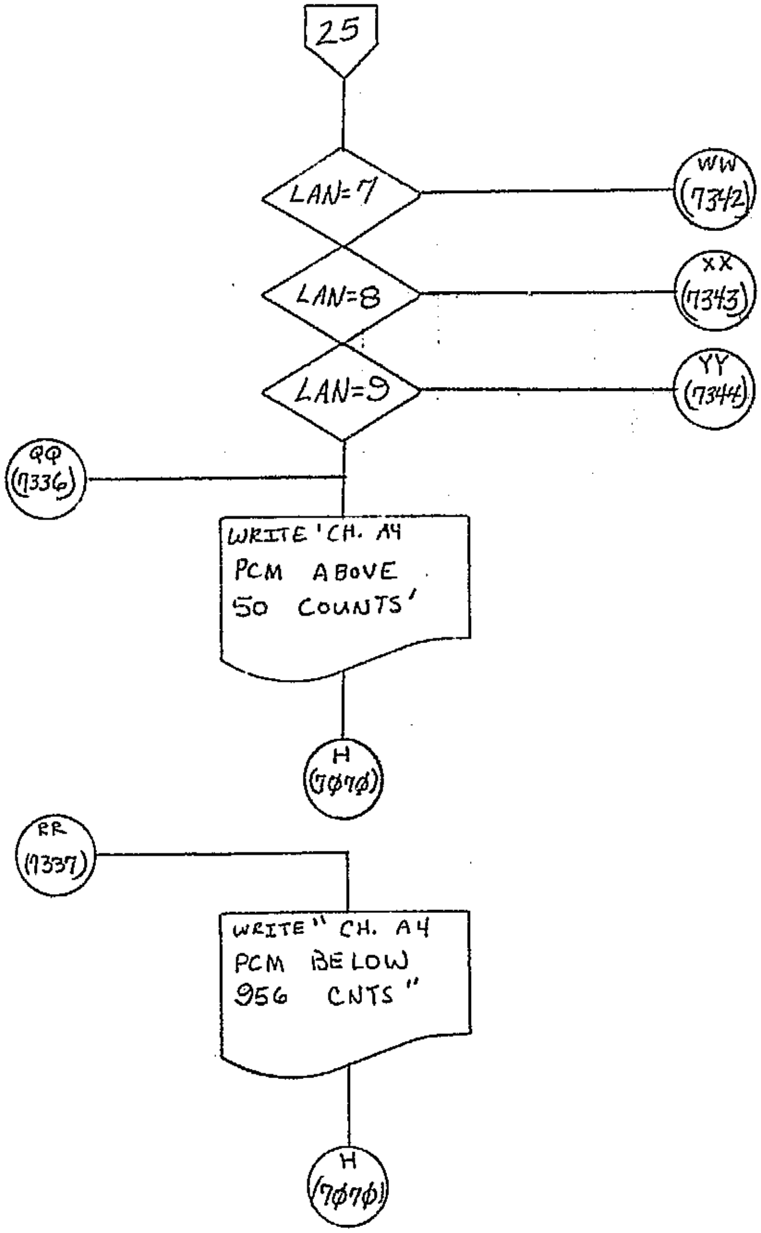

20
STOHMS
CONVERT HEATED CAL START TIME TO HOURS, MINUTE & SECONDS.
WRITE 'HEATED RESPONSIVITY ABORTED'
INCREMENT LINE COUNTER: LINE=LINE+1
LL (7322)
LIL= LSWL-1
FRTIME = SSTIME (1,LIL)
STOHMS
CONVERT SWL CAL START TIME TO HOURS, MINUTES, & SECONDS
WRITE 'SWL RESPONSIVITY ABORTED'
21

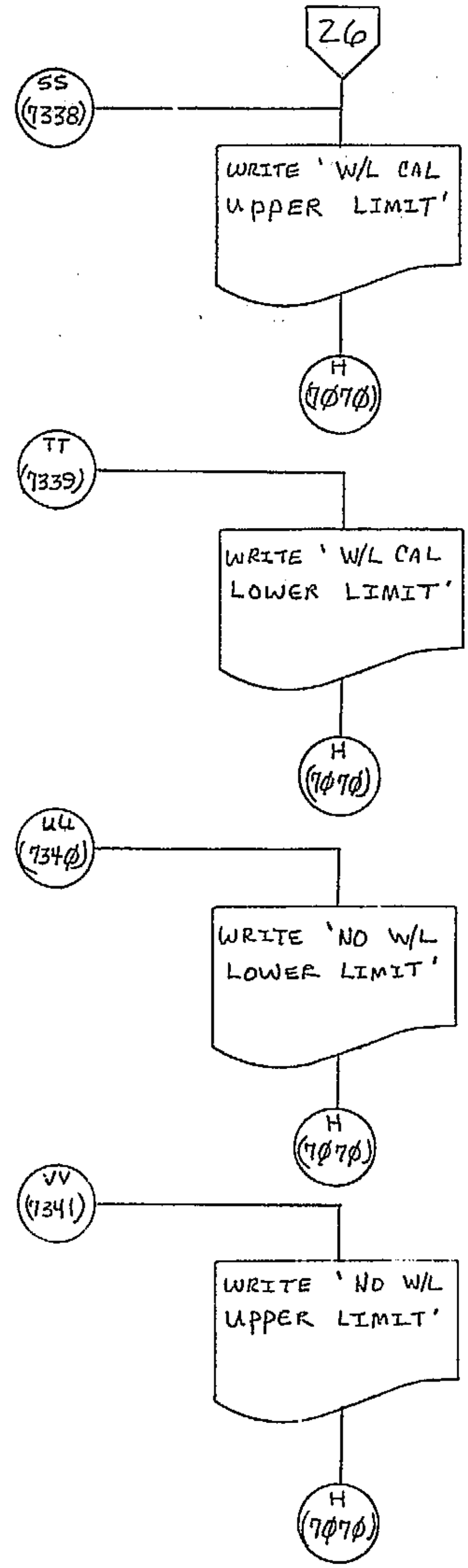

21
LZNE=LZNE+1
FF
(7330)
TANCTR(JPAR) =
TANCTR(JPAR) + 1
IS
SAMPLE COUNTER
LESS THAN
576?
Y
N
NSAM=NSAM-576
IS
SAMPLE COUNT
.GT. 384
?
Y
MM
(7331)
N
IS
SAMPLE COUNT
.GT. 192
?
Y
NN
(7332)
N
LL = 3
OO
(7333)

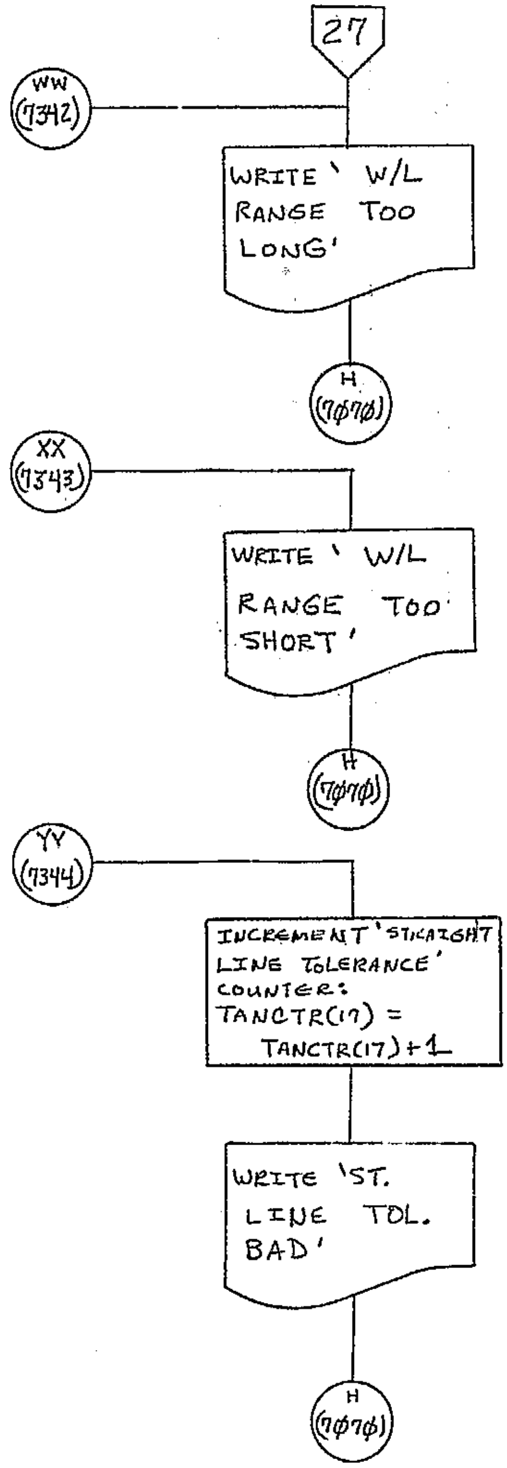

22
MM (7331)
LL=5
OD (7333)
NN (7332)
LL = 4
OO (7333)
COMPUTE RECORD NUMBER (IREC):
IREC= FRSTR(1) − LL
AMOV
MOVE FRAME TIME TO D.P. FIRSTR
AMOV
MOVE FRAME DELTA TIME TO D.P. DLT.
COMPUTE PARAMETER TIME:
FRTIME= FIRSTR + (DLT *(NSAM-1))
23

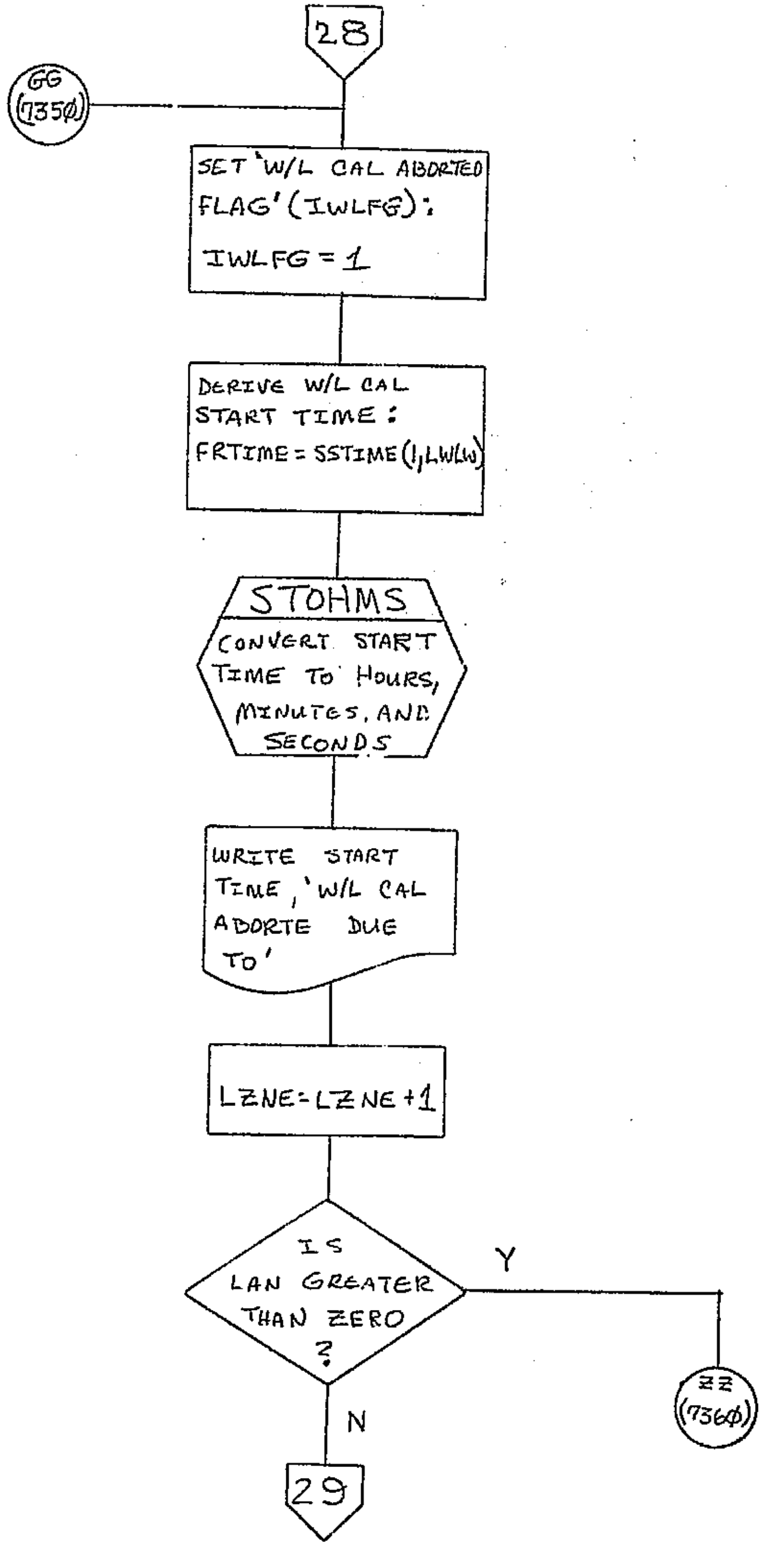

23
STOHMS
CONVERT FRTIME TO HOURS, MINUTES, AND SECONDS.
RFMT(9) = RPAR(18)
IS ANOMALY CAUSED BY ERROR OTHER THAN WRONG SAMPLE ?
Y
PP (7335)
N
RFMT(14)=RINT(1)
RFMT(15)=RINT(2)
WRITE 'SCAN REJECTED' ANOMALY.
LZNE LZNE+1
H (7070)

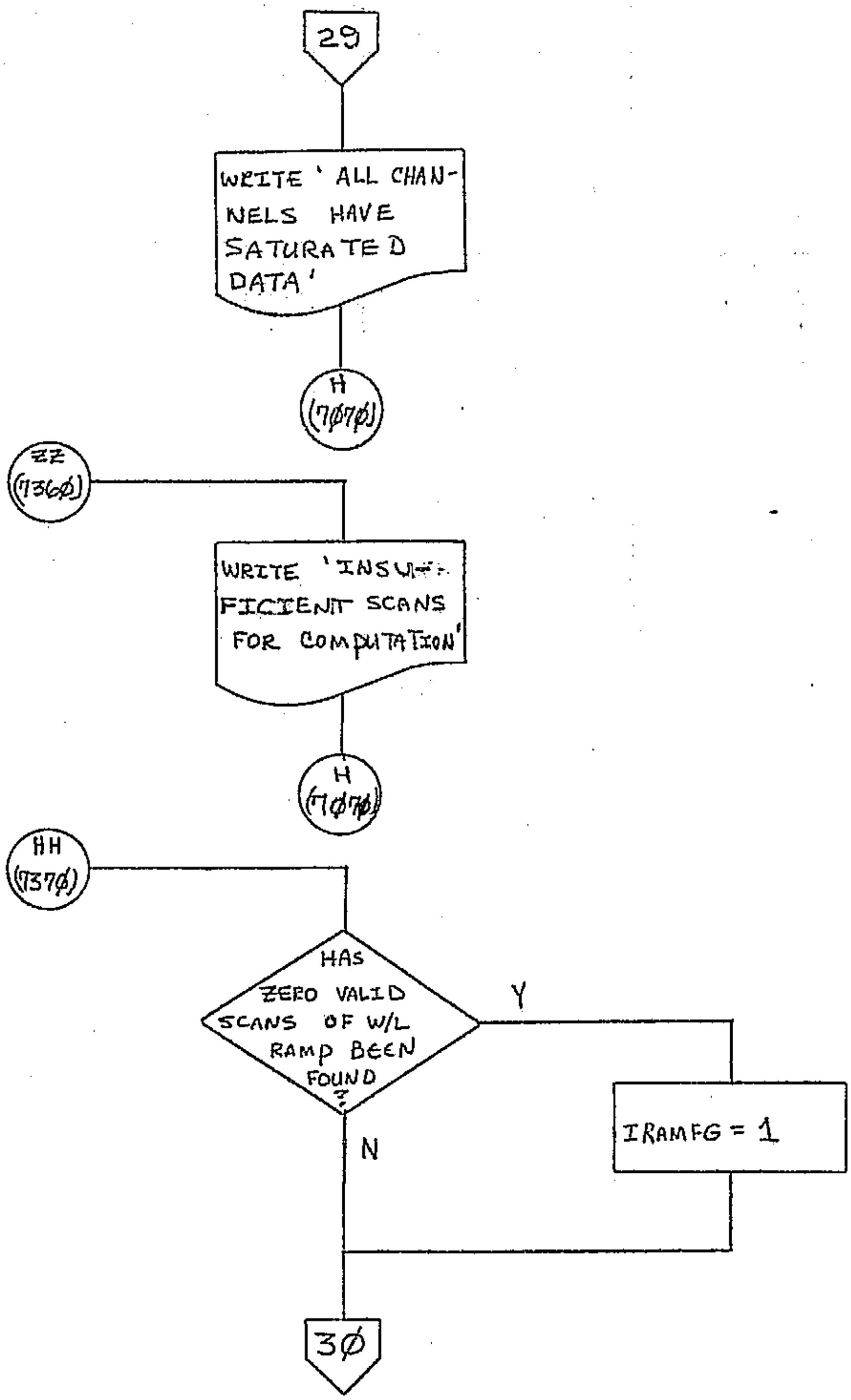

24
PP
(7335)
RFMT(14) = RBLK(1)
RFMT(15) = RBLK(2)
WRITE ' SCAN
REJECTED '
ANOMALY.
LZNE = LZNE + 1
COMPUTE GOTO:
LAN = ?
LAN = 1
QQ
(7336)
LAN = 2
RR
(7337)
LAN = 3
SS
(7338)
LAN = 4
TT
(7339)
LAN = 5
UU
(7340)
LAN = 6
VV
(7341)
25

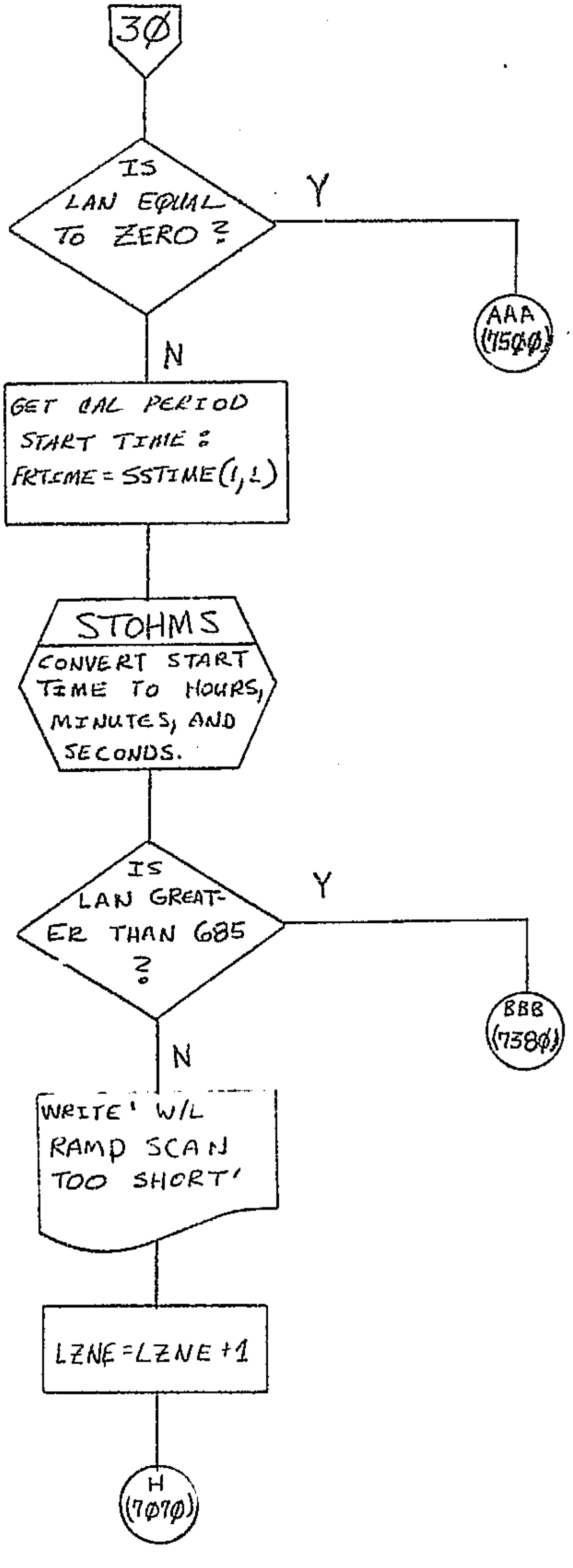

25
LAN=7
WW (7342)
LAN=8
XX (7343)
LAN=9
YY (7344)
QQ (7336)
WRITE 'CH. A4 PCM ABOVE 50 COUNTS'
H (7Ø7Ø)
RR (7337)
WRITE "CH. A4 PCM BELOW 956 CNTS"
H (7Ø7Ø)

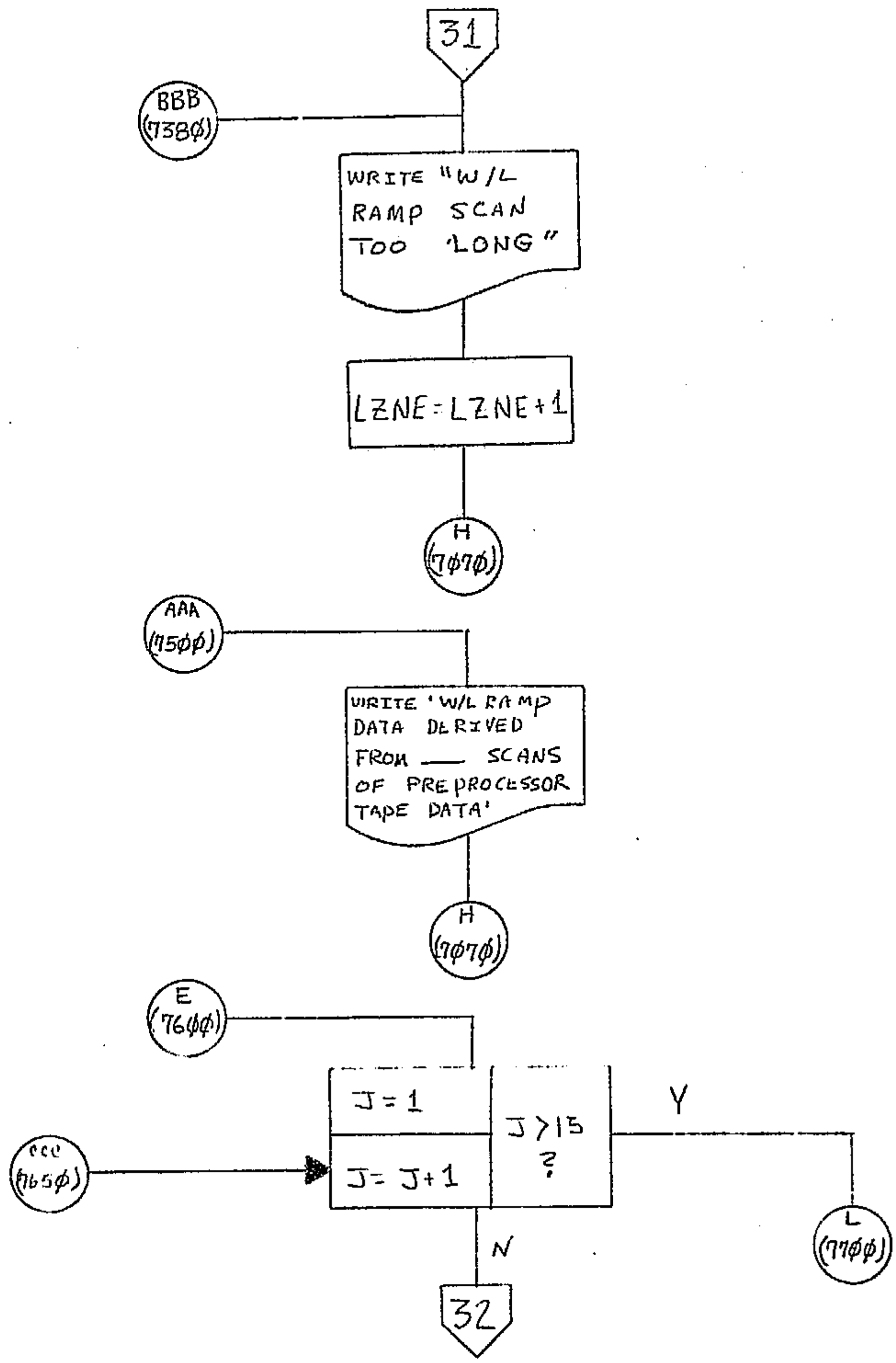

26
SS (7338)
WRITE 'W/L CAL UPPER LIMIT'
H (7Ø7Ø)
TT (7339)
WRITE 'W/L CAL LOWER LIMIT'
H (7Ø7Ø)
UU (734Ø)
WRITE 'NO W/L LOWER LIMIT'
H (7Ø7Ø)
VV (7341)
WRITE 'NO W/L UPPER LIMIT'
H (7Ø7Ø)

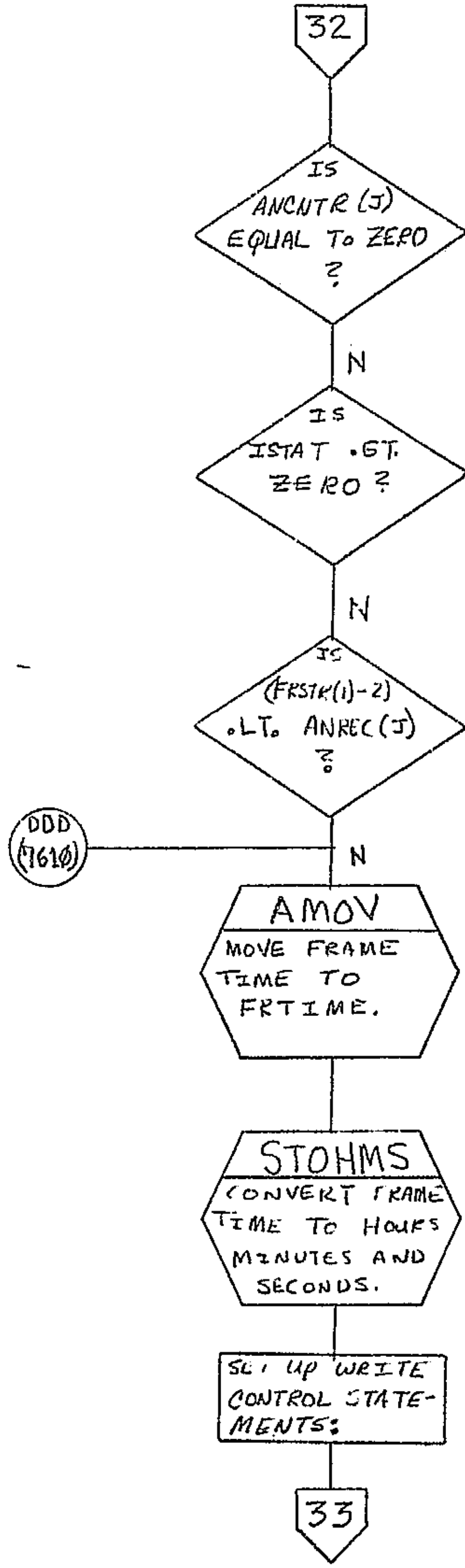

27
WW (7342)
WRITE ' W/L RANGE TOO LONG'
H (7Ø7Ø)
XX (7343)
WRITE ' W/L RANGE TOO SHORT'
H (7Ø7Ø)
YY (7344)
INCREMENT 'STRAIGHT LINE TOLERANCE' COUNTER: TANCTR(17) = TANCTR(17)+1
WRITE 'ST. LINE TOL. BAD'
H (7Ø7Ø)

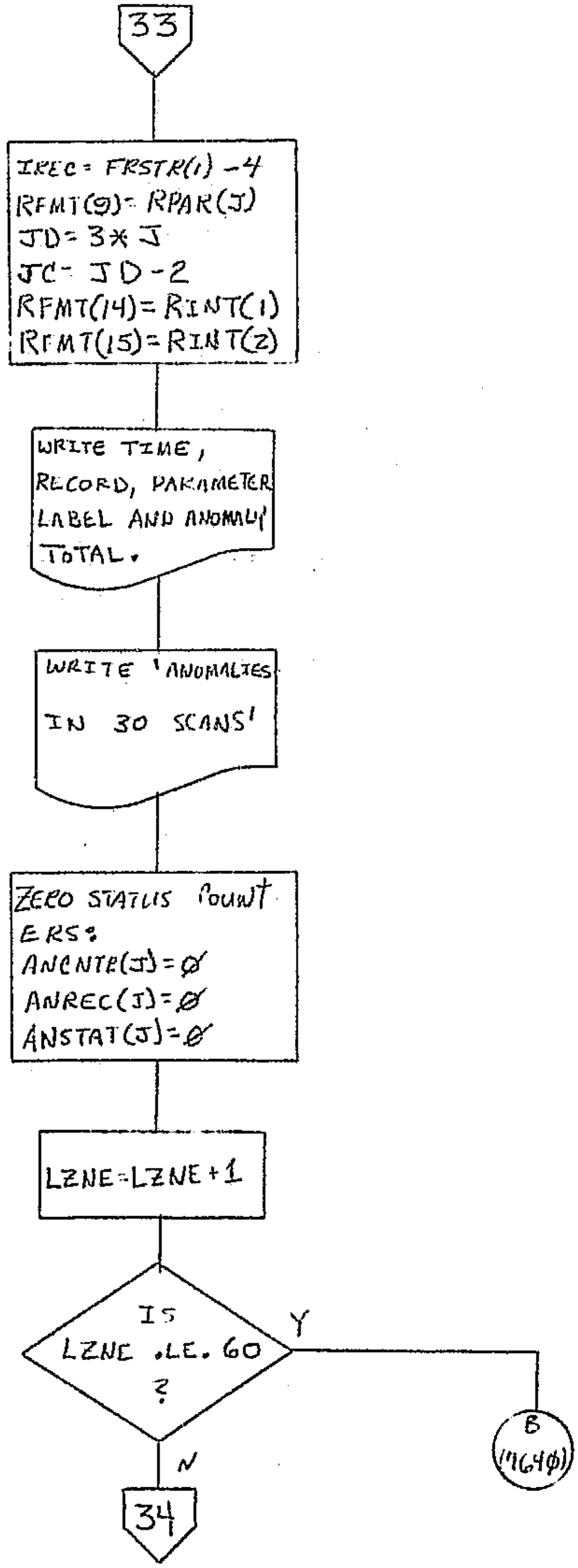

28
GG (735Ø)
SET 'W/L CAL ABORTED FLAG' (IWLFG):
IWLFG = 1
DERIVE W/L CAL START TIME:
FRTIME = SSTIME (1,LWLW)
STOHMS
CONVERT START TIME TO HOURS, MINUTES, AND SECONDS
WRITE START TIME, 'W/L CAL ABORTE DUE TO'
LZNE=LZNE+1
IS LAN GREATER THAN ZERO ?
Y
ZZ (736Ø)
N
29

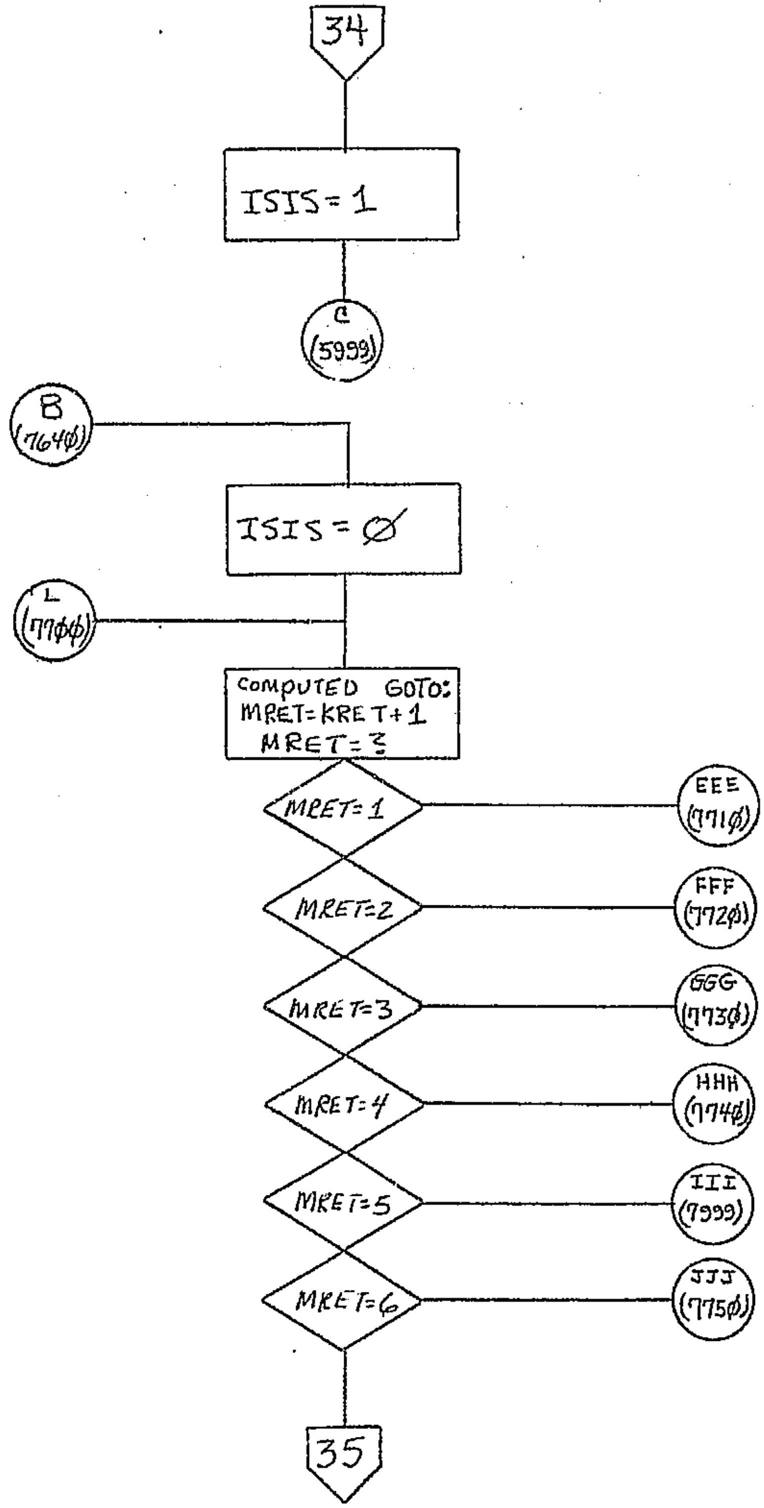

29
WRITE 'ALL CHANNELS HAVE SATURATED DATA'
H (7Ø7Ø)
ZZ (736Ø)
WRITE 'INSUFFICIENT SCANS FOR COMPUTATION'
H (7Ø7Ø)
HH (737Ø)
HAS ZERO VALID SCANS OF W/L RAMP BEEN FOUND ?
Y
N
IRAMFG = 1
3Ø

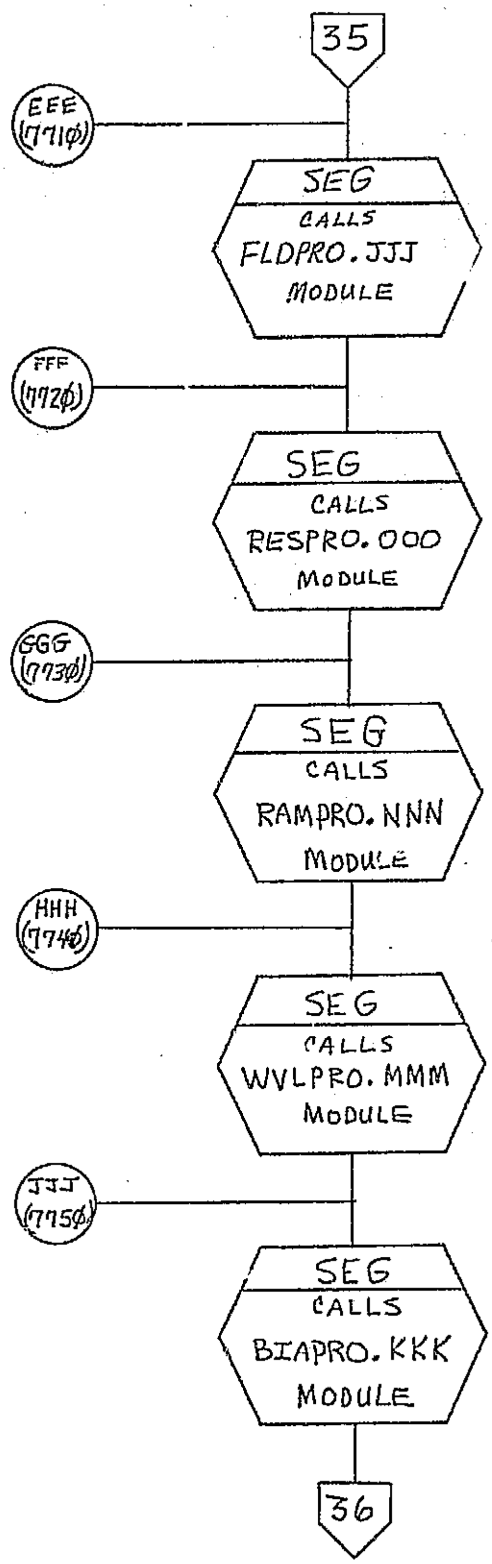

30
IS LAN EQUAL TO ZERO ?
Y
AAA (7500)
N
GET CAL PERIOD START TIME : FRTIME = SSTIME(1,L)
STOHMS
CONVERT START TIME TO HOURS, MINUTES, AND SECONDS.
IS LAN GREATER THAN 685 ?
Y
BBB (7380)
N
WRITE 'W/L RAMP SCAN TOO SHORT'
LZNE=LZNE+1
H (7070)

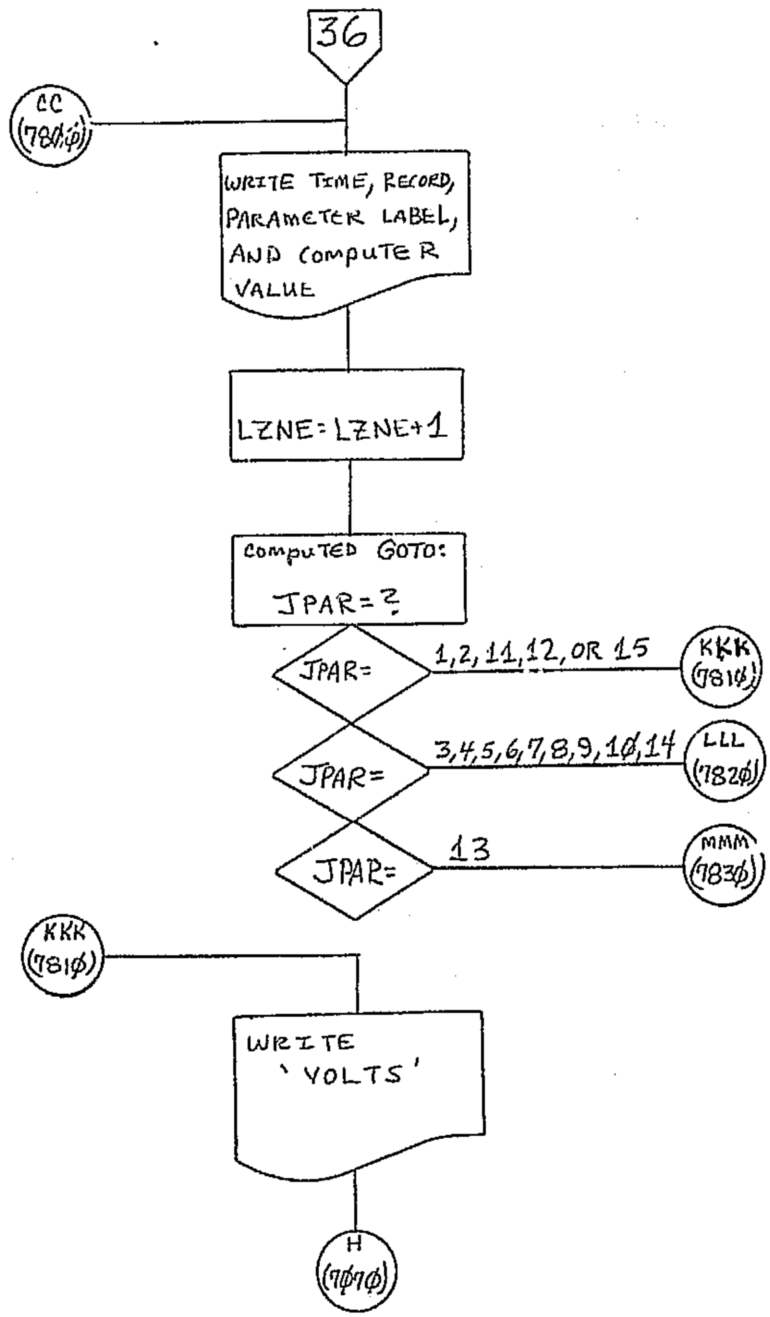

31
BBB
(738Ø)
WRITE "W/L
RAMP SCAN
TOO LONG"
LZNE=LZNE+1
H
(7Ø7Ø)
AAA
(75ØØ)
WRITE 'W/L RAMP
DATA DERIVED
FROM ___ SCANS
OF PREPROCESSOR
TAPE DATA'
H
(7Ø7Ø)
E
(76ØØ)
J = 1
J = J+1
J > 15
?
Y
CCC
(765Ø)
N
32
L
(77ØØ)

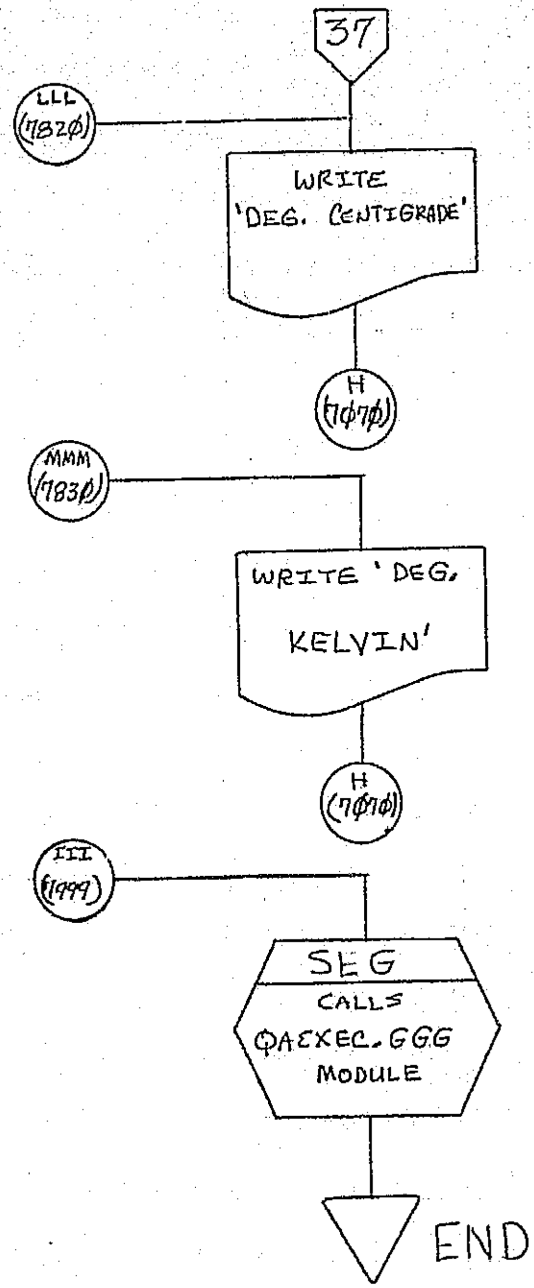

32
IS ANCNTR (J) EQUAL TO ZERO ?
Y
CCC (765Ø)
N
IS ISTAT .GT. ZERO ?
Y
DDD (761Ø)
N
IS (FRSTR(1)-2) .LT. ANREC (J) ?
Y
CCC (765Ø)
DDD (761Ø)
N
AMOV
MOVE FRAME TIME TO FRTIME.
STOHMS
CONVERT FRAME TIME TO HOURS MINUTES AND SECONDS.
SET UP WRITE CONTROL STATEMENTS:
33

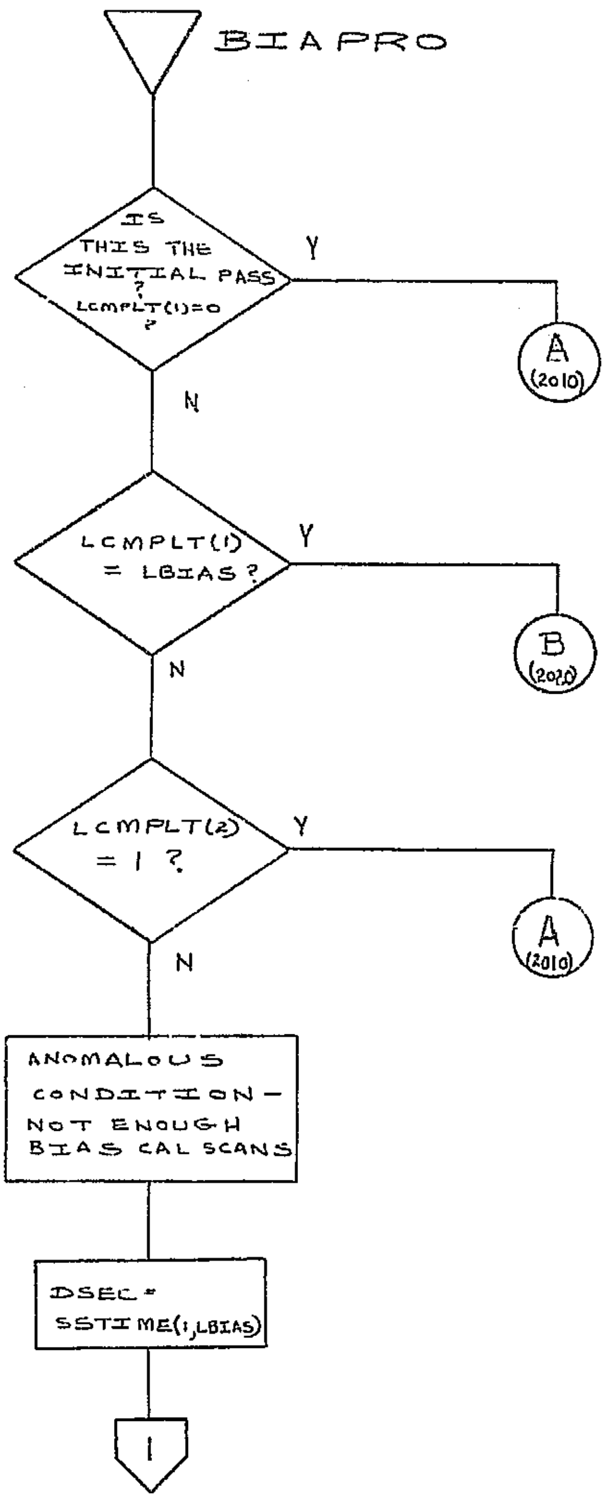

33
IREC = FRSTR(I) -4
RFMT(9) = RPAR(J)
JD = 3*J
JC = JD-2
RFMT(14) = RINT(1)
RFMT(15) = RINT(2)
WRITE TIME, RECORD, PARAMETER LABEL AND ANOMALY TOTAL.
WRITE 'ANOMALIES IN 30 SCANS'
ZERO STATUS COUNTERS:
ANCNTR(J) = Ø
ANREC(J) = Ø
ANSTAT(J) = Ø
LZNE = LZNE + 1
IS LZNE .LE. 60 ?
Y
N
B
(1764Ø)
34

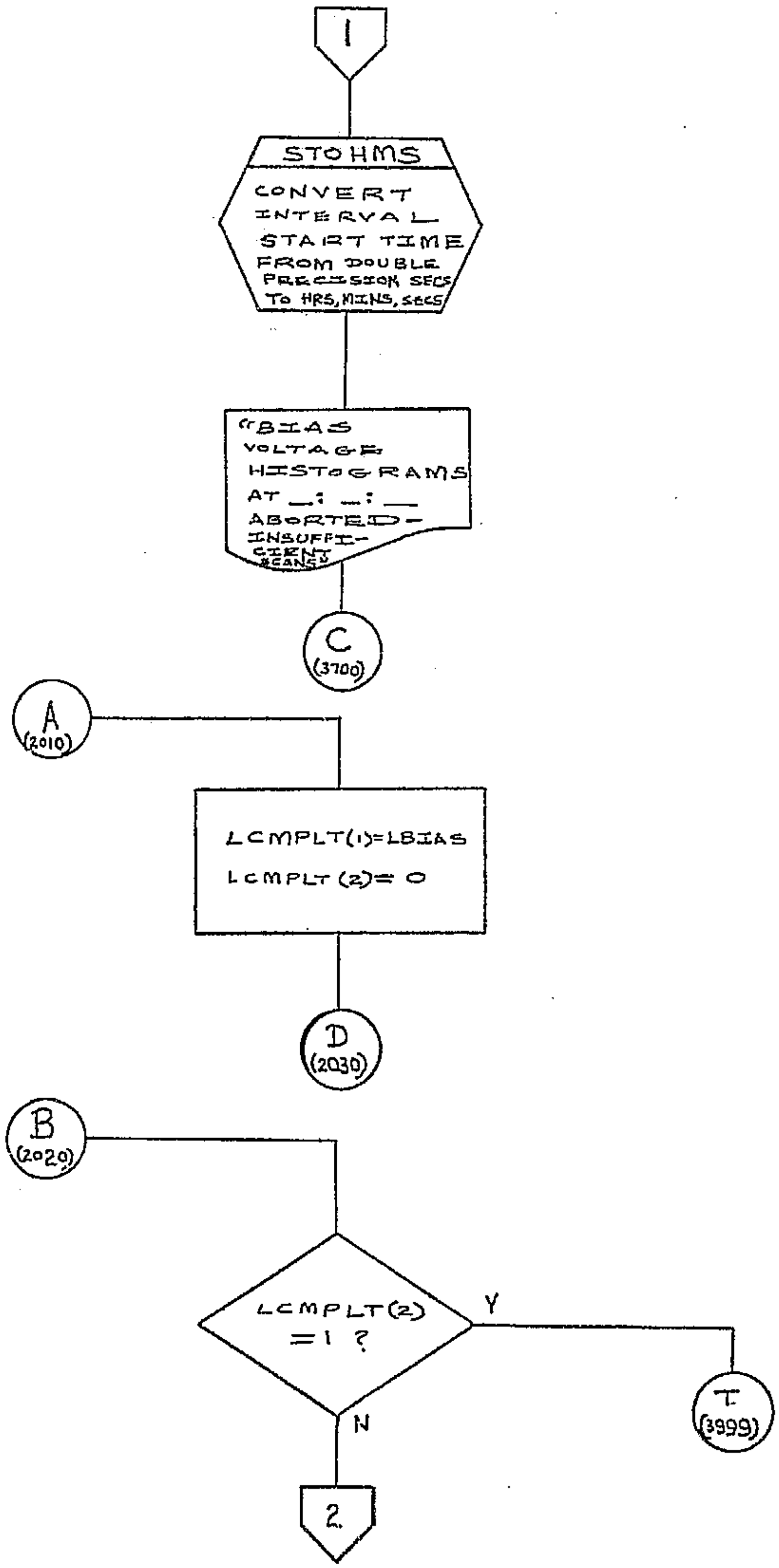

34
ISIS = 1
C (5999)
B (7640)
ISIS = 0
L (7700)
COMPUTED GOTO:
MRET=KRET+1
MRET=3
MRET=1
EEE (7710)
MRET=2
FFF (7720)
MRET=3
GGG (7730)
MRET=4
HHH (7740)
MRET=5
III (7999)
MRET=6
JJJ (7750)
35

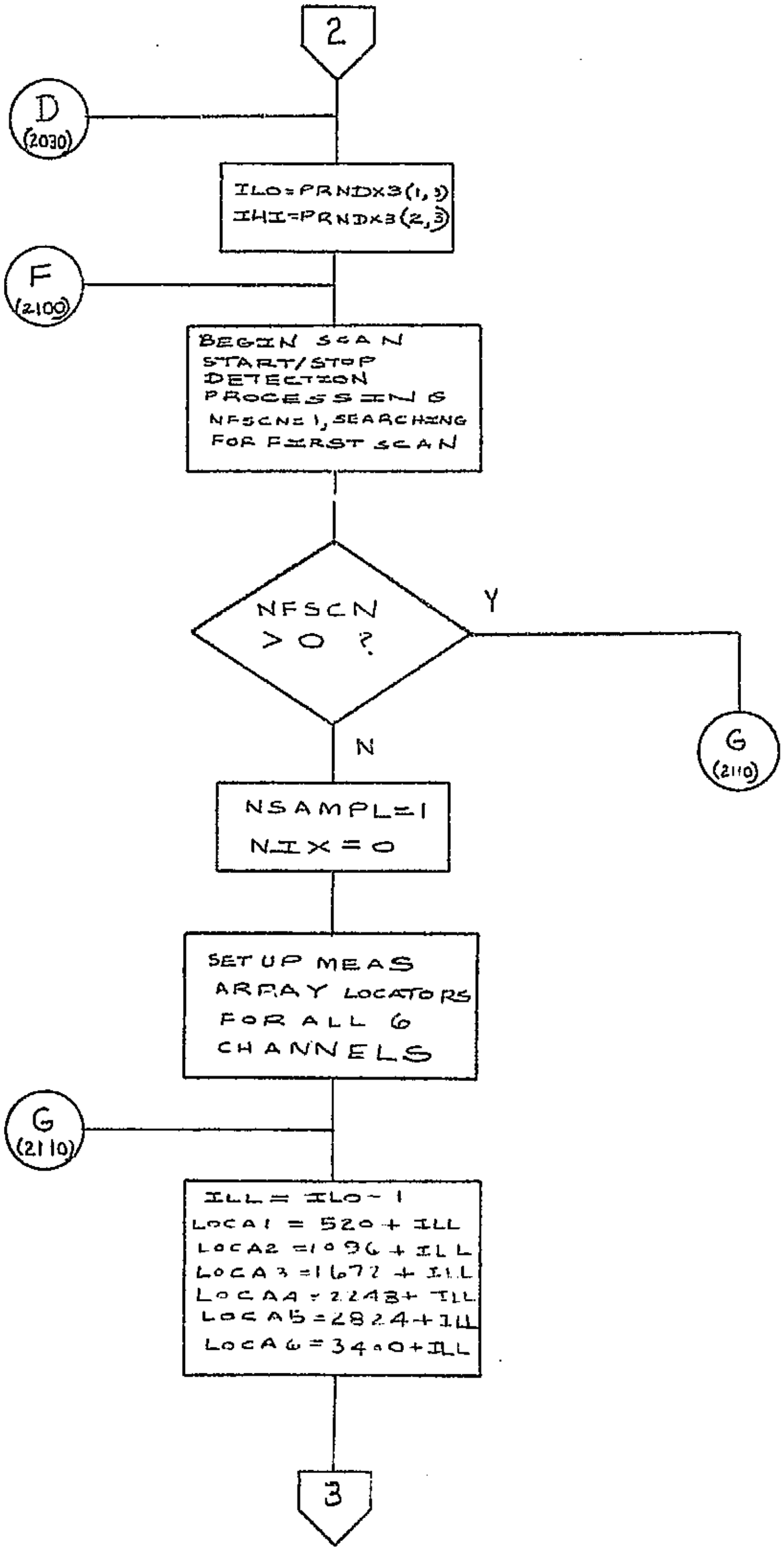

35
EEE (771Ø)
SEG
CALLS
FLDPRO.JJJ
MODULE
FFF (772Ø)
SEG
CALLS
RESPRO.OOO
MODULE
GGG (773Ø)
SEG
CALLS
RAMPRO.NNN
MODULE
HHH (774Ø)
SEG
CALLS
WVLPRO.MMM
MODULE
JJJ (775Ø)
SEG
CALLS
BIAPRO.KKK
MODULE
36

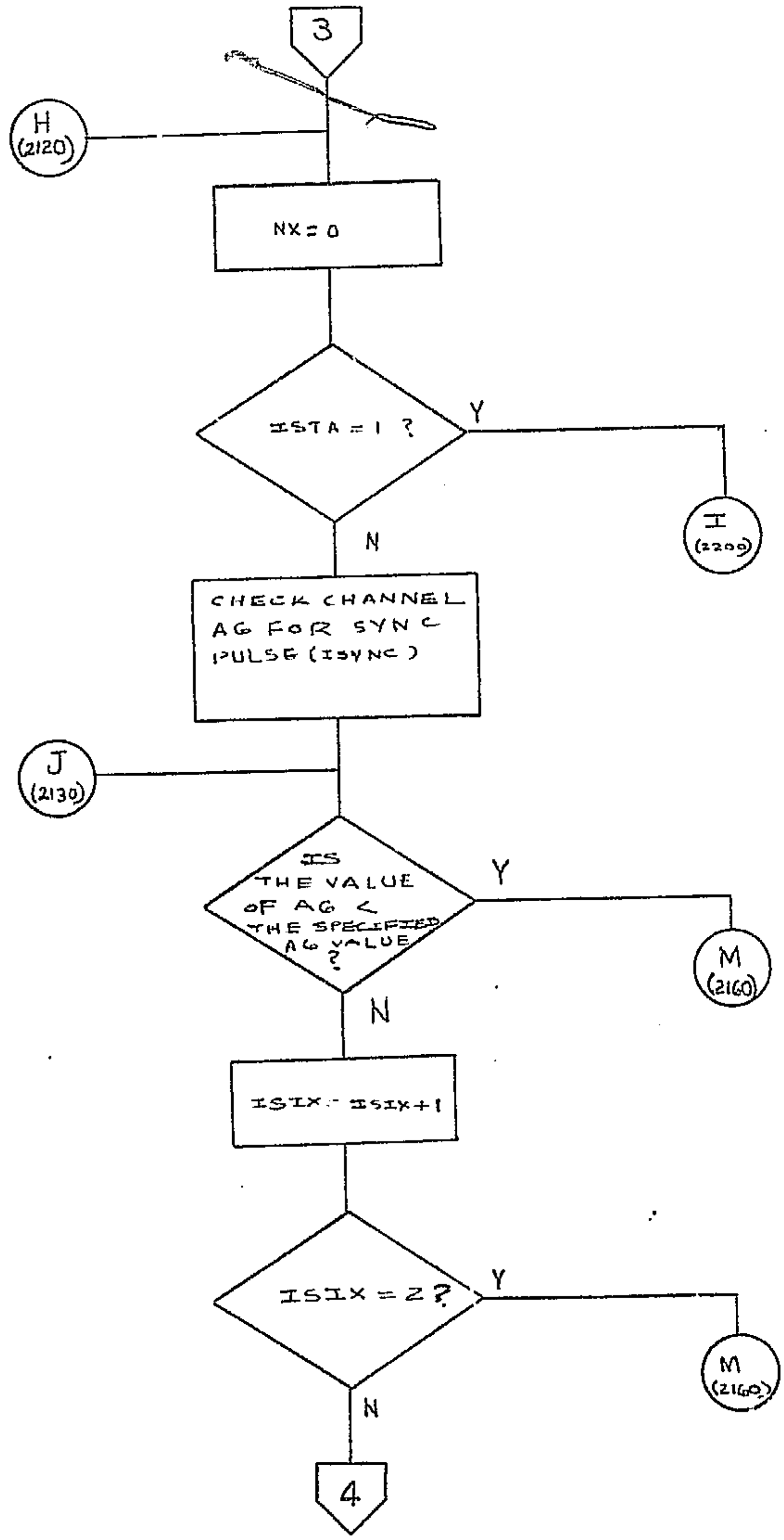

36
CC
(78ØØ)
WRITE TIME, RECORD, PARAMETER LABEL, AND COMPUTER VALUE
LZNE=LZNE+1
COMPUTED GOTO:
JPAR=?
JPAR=
1,2,11,12, OR 15
KKK
(781Ø)
JPAR=
3,4,5,6,7,8,9,1Ø,14
LLL
(782Ø)
JPAR=
13
MMM
(783Ø)
KKK
(781Ø)
WRITE 'VOLTS'
H
(7Ø7Ø)

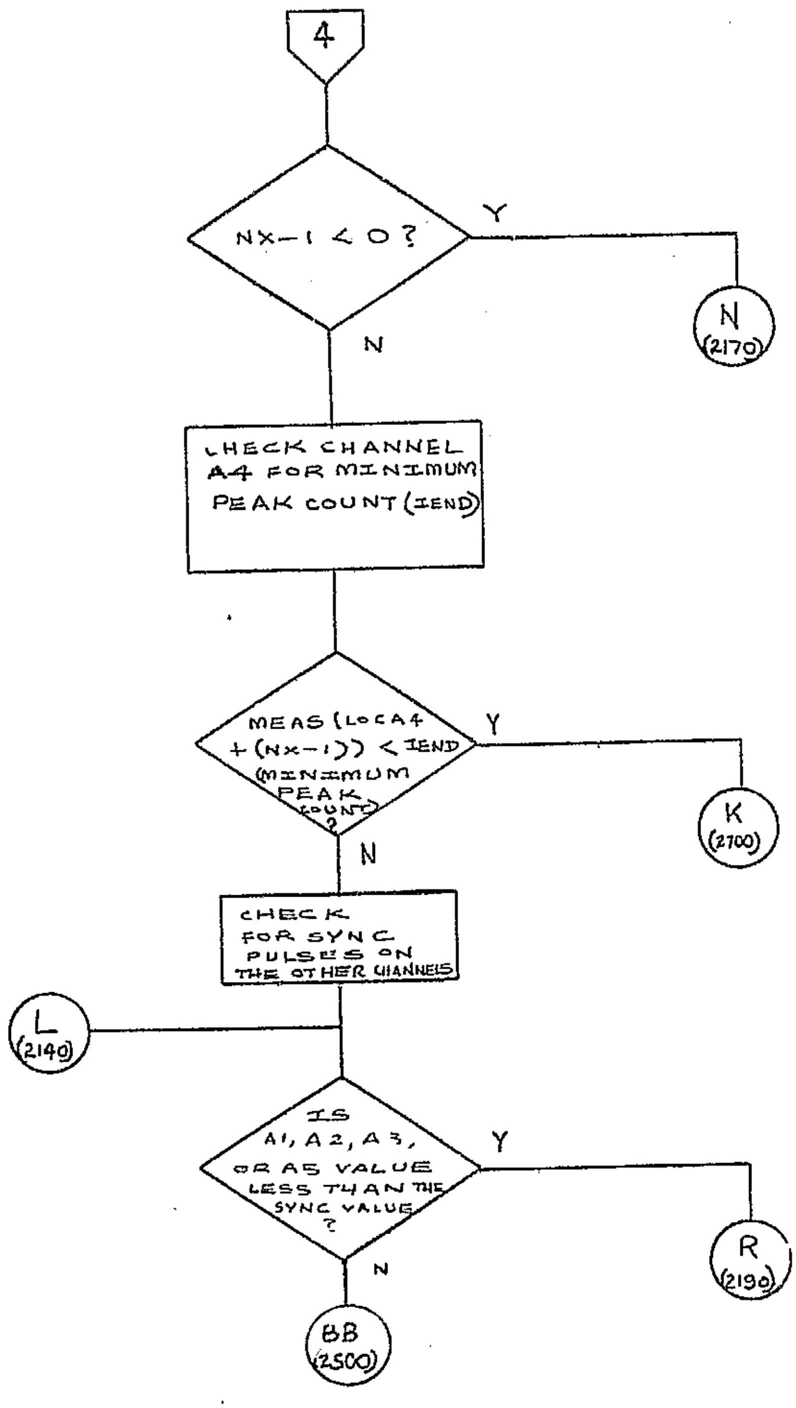

37
LLL
(782ϕ)
WRITE
'DEG. CENTIGRADE'
H
(7ϕ7ϕ)
MMM
(783ϕ)
WRITE 'DEG.
KELVIN'
H
(7ϕ7ϕ)
III
(1999)
SEG
CALLS
ΦAEXEC.GGG
MODULE
END

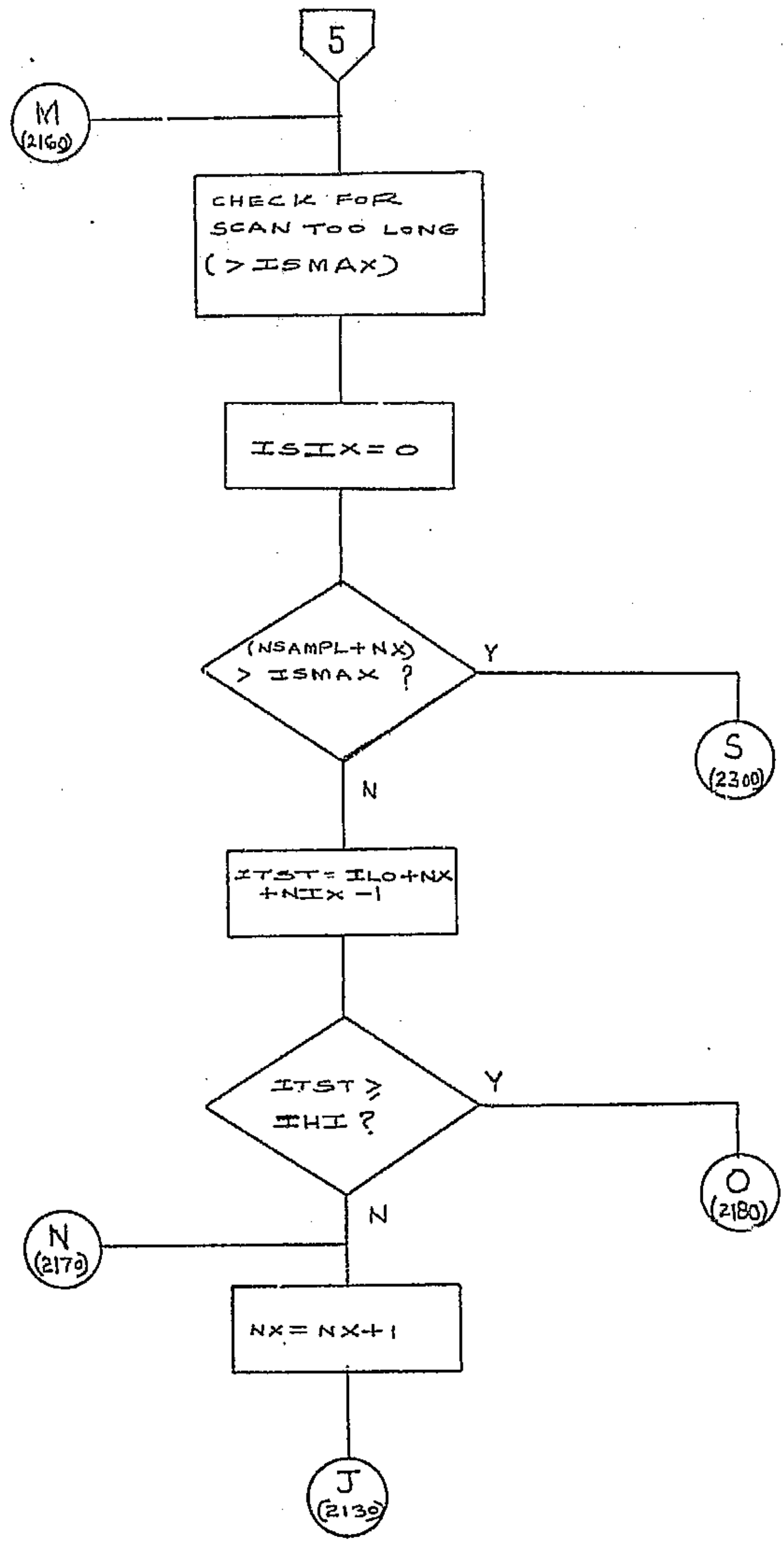

BIAPRO
IS THIS THE INITIAL PASS ? LCMPLT(1)=0 ?
Y
N
A (2010)
LCMPLT(1) = LBIAS ?
Y
N
B (2020)
LCMPLT(2) = 1 ?
Y
N
A (2010)
ANOMALOUS CONDITION — NOT ENOUGH BIAS CAL SCANS
DSEC = SSTIME(1,LBIAS)
1

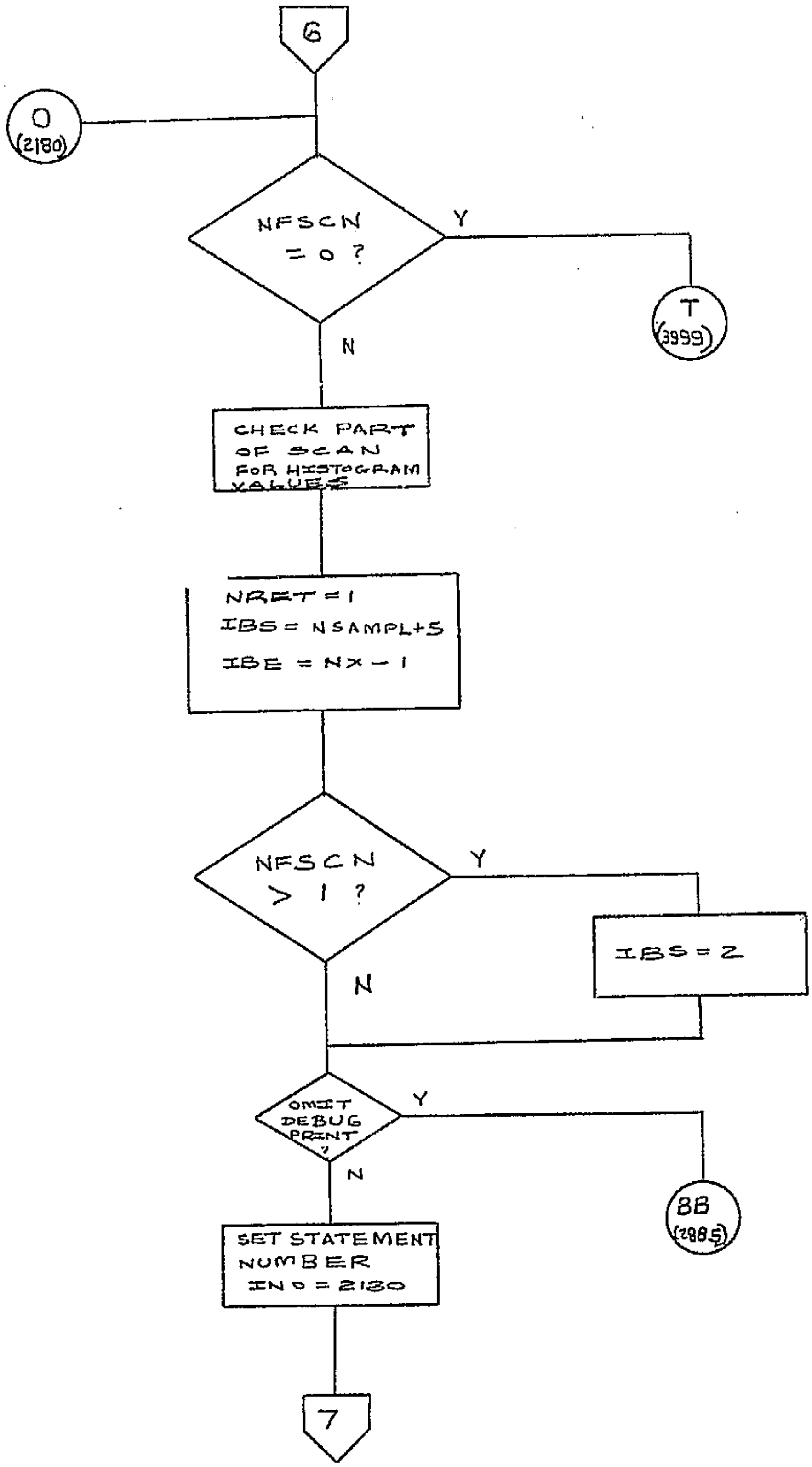

1
STOHMS
CONVERT INTERVAL START TIME FROM DOUBLE PRECISION SECS TO HRS, MINS, SECS
"BIAS VOLTAGE HISTOGRAMS AT _: _: __ ABORTED- INSUFFI- CIENT SCANS"
C (3700)
A (2010)
LCMPLT(1)=LBIAS
LCMPLT(2)= 0
D (2030)
B (2020)
LCMPLT(2) =1 ?
Y
N
T (3999)
2

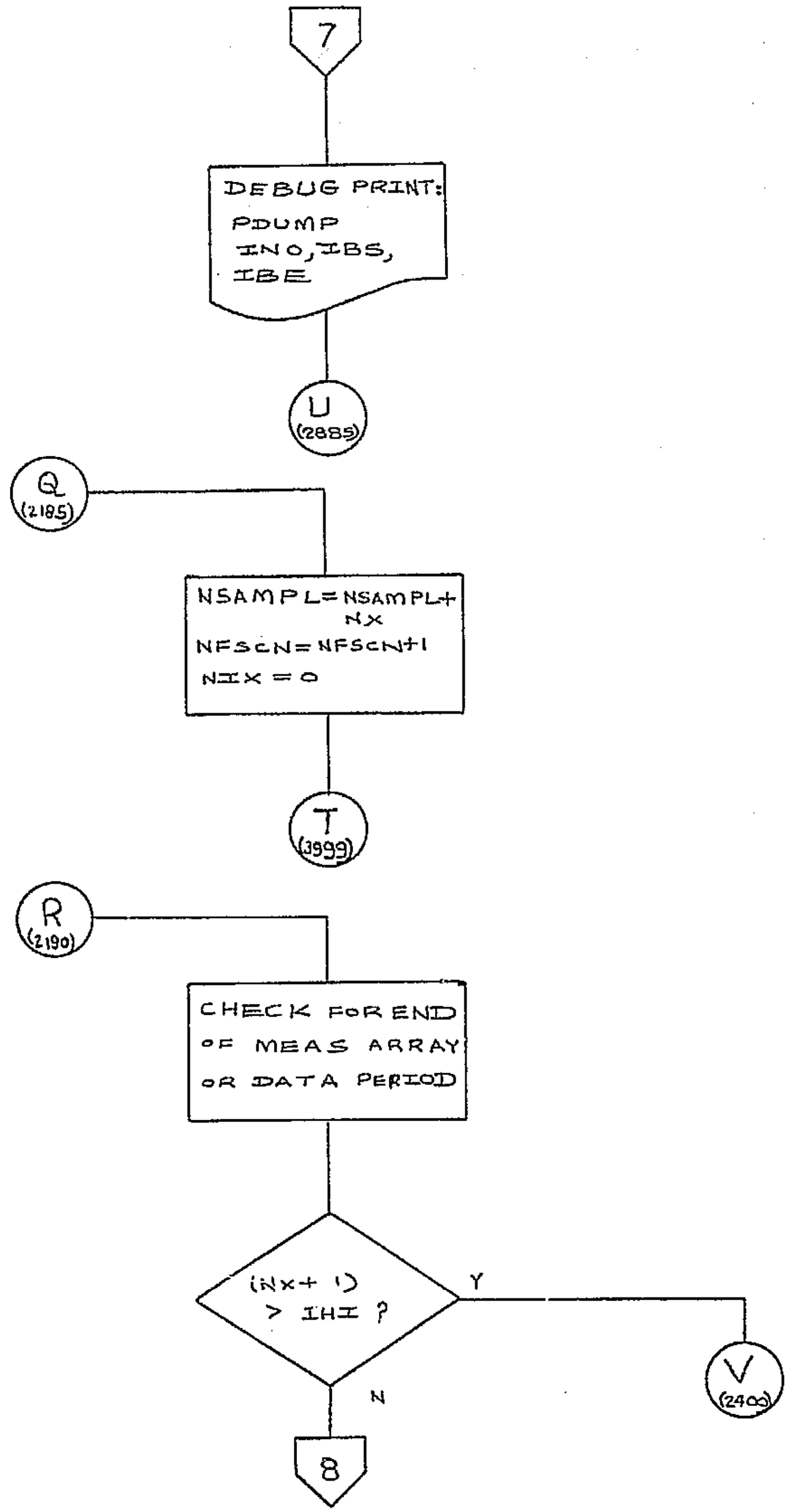

2
D
(2030)
ILO=PRNDX3(1,3)
IHI=PRNDX3(2,3)
F
(2100)
BEGIN SCAN START/STOP DETECTION PROCESSING
NFSCN=1, SEARCHING FOR FIRST SCAN
NFSCN >0 ?
Y
G
(2110)
N
NSAMPL=1
NIX=0
SET UP MEAS ARRAY LOCATORS FOR ALL 6 CHANNELS
G
(2110)
ILL= ILO-1
LOCA1 = 520+ILL
LOCA2 =1096+ILL
LOCA3 =1672+ILL
LOCA4 =2248+ILL
LOCA5 =2824+ILL
LOCA6 =3400+ILL
3

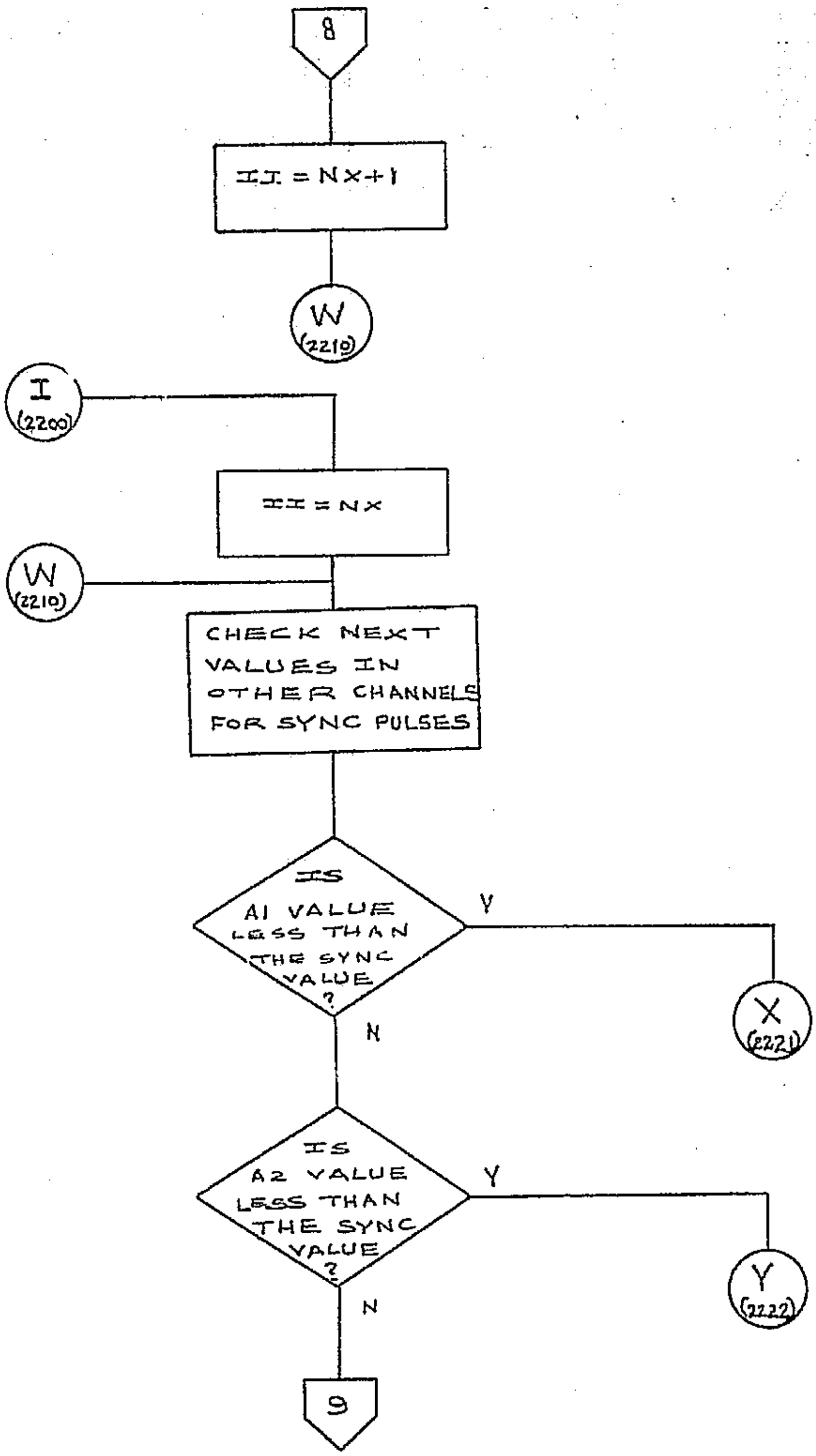

3
H
(2120)
NX = 0
ISTA = 1 ?
Y
I
(2209)
N
CHECK CHANNEL AG FOR SYNC PULSE (ISYNC)
J
(2130)
IS THE VALUE OF AG < THE SPECIFIED AG VALUE ?
Y
M
(2160)
N
ISIX = ISIX+1
ISIX = 2?
Y
M
(2160)
N
4

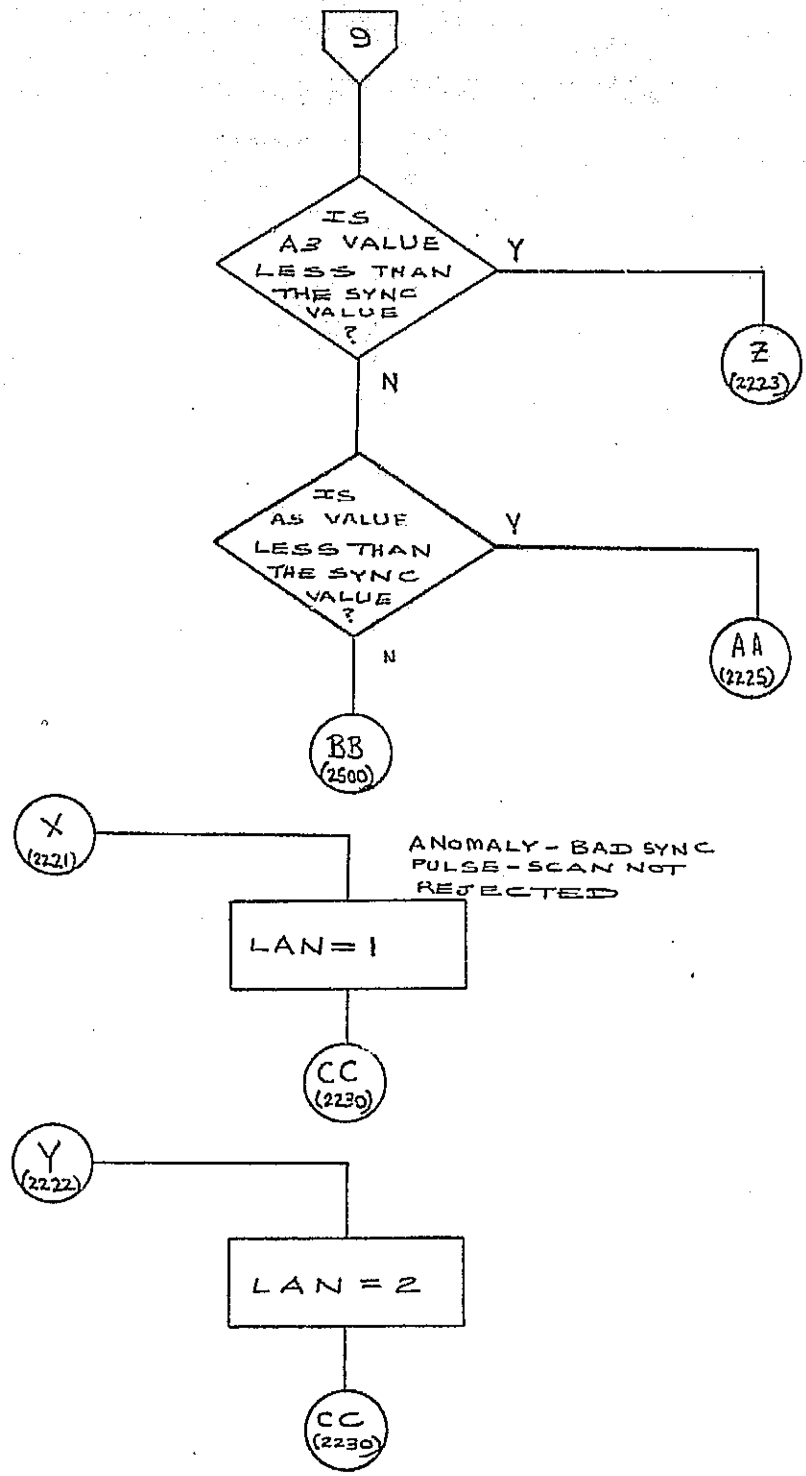

4
NX-1 < 0?
Y
N
N (2170)
CHECK CHANNEL A4 FOR MINIMUM PEAK COUNT (IEND)
MEAS (LOCA4 + (NX-1)) < IEND (MINIMUM PEAK COUNT) ?
Y
N
K (2700)
CHECK FOR SYNC PULSES ON THE OTHER CHANNELS
L (2140)
IS A1, A2, A3, OR A5 VALUE LESS THAN THE SYNC VALUE ?
Y
N
R (2190)
BB (2500)

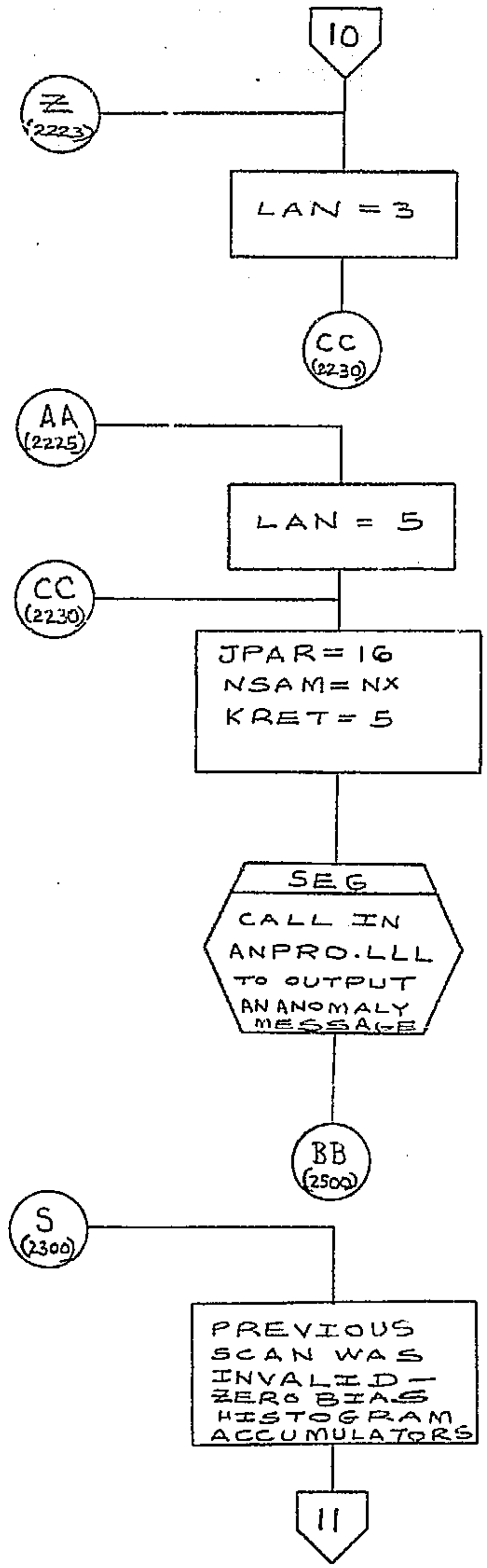

5
M
(2160)
CHECK FOR
SCAN TOO LONG
(>ISMAX)
ISIX=0
(NSAMPL+NX)
> ISMAX ?
Y
S
(2300)
N
ITST=ILO+NX
+NIX -1
ITST ≥
IHI ?
Y
O
(2180)
N
N
(2170)
NX=NX+1
J
(2130)

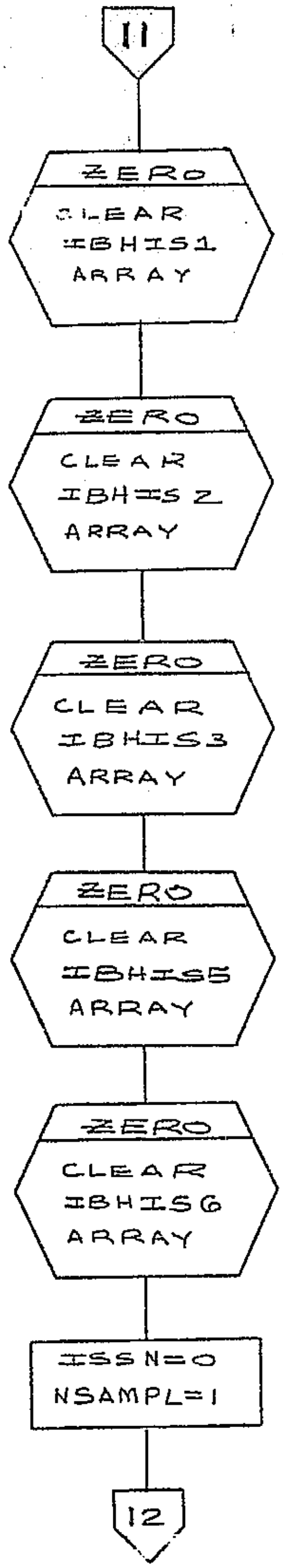

6
O
(2180)
NFSCN = 0 ?
Y
N
T
(3999)
CHECK PART OF SCAN FOR HISTOGRAM VALUES
NRET = 1
IBS = NSAMPL+5
IBE = NX - 1
NFSCN > 1 ?
Y
N
IBS = 2
OMIT DEBUG PRINT ?
Y
N
BB
(2985)
SET STATEMENT NUMBER IND = 2180
7

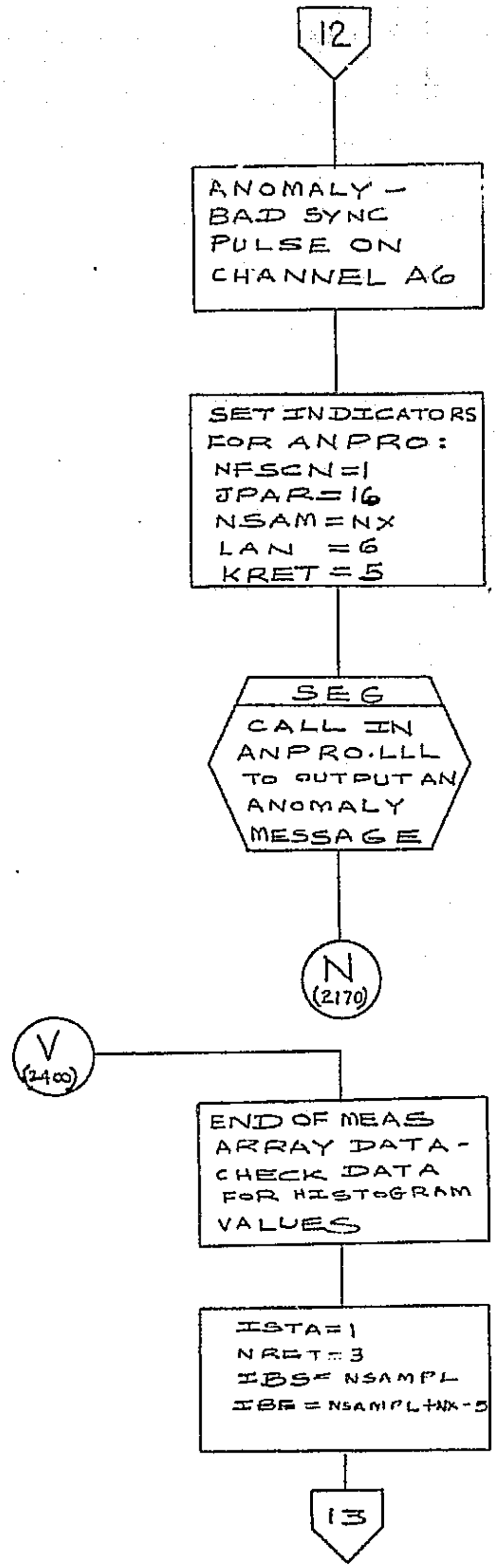

7
DEBUG PRINT:
PDUMP
INO, IBS,
IBE
U
(2885)
Q
(2185)
NSAMPL=NSAMPL+
NX
NFSCN=NFSCN+1
NIX = 0
T
(3999)
R
(2190)
CHECK FOR END
OF MEAS ARRAY
OR DATA PERIOD
(NX+ 1)
> IHI ?
Y
N
V
(2400)
8

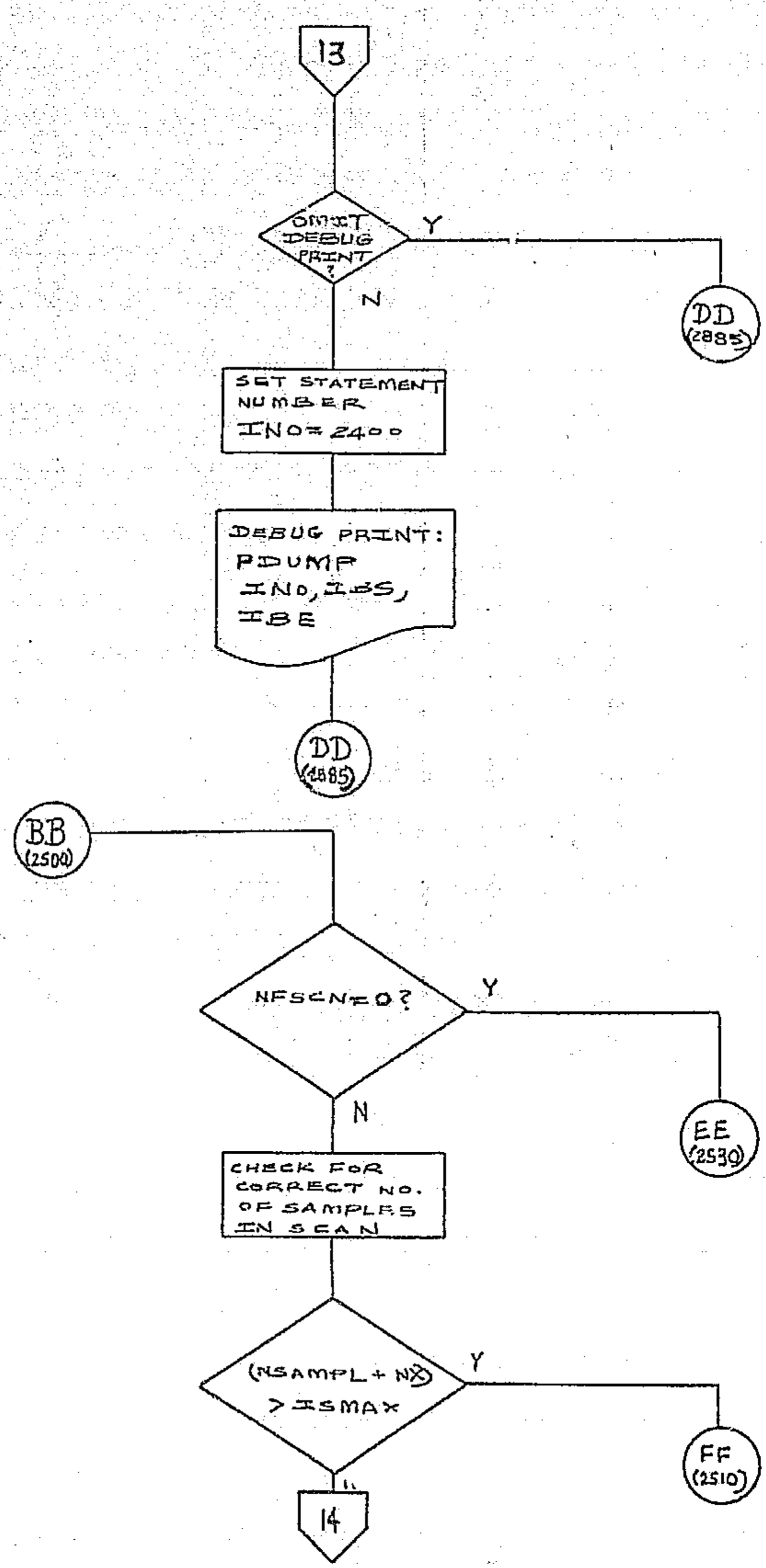

8
II = NX+1
W
(2210)
I
(2200)
II = NX
W
(2210)
CHECK NEXT
VALUES IN
OTHER CHANNELS
FOR SYNC PULSES
IS
A1 VALUE
LESS THAN
THE SYNC
VALUE
?
Y
N
X
(2221)
IS
A2 VALUE
LESS THAN
THE SYNC
VALUE
?
Y
N
Y
(2222)
9

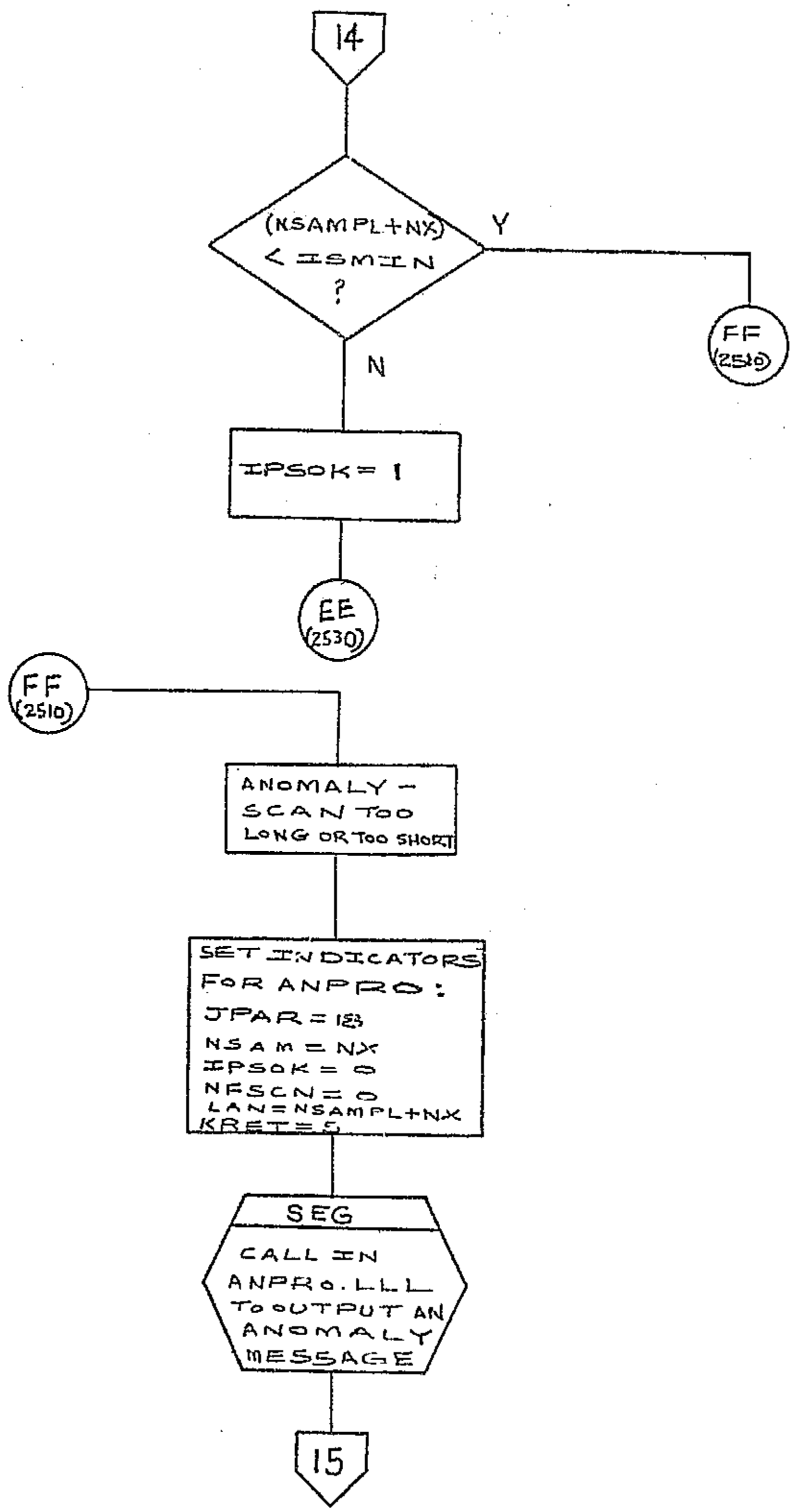

9
IS
A3 VALUE
LESS THAN
THE SYNC
VALUE
?
Y
N
Z
(2223)
IS
A5 VALUE
LESS THAN
THE SYNC
VALUE
?
Y
N
AA
(2225)
BB
(2500)
X
(2221)
ANOMALY - BAD SYNC
PULSE - SCAN NOT
REJECTED
LAN = 1
CC
(2230)
Y
(2222)
LAN = 2
CC
(2230)

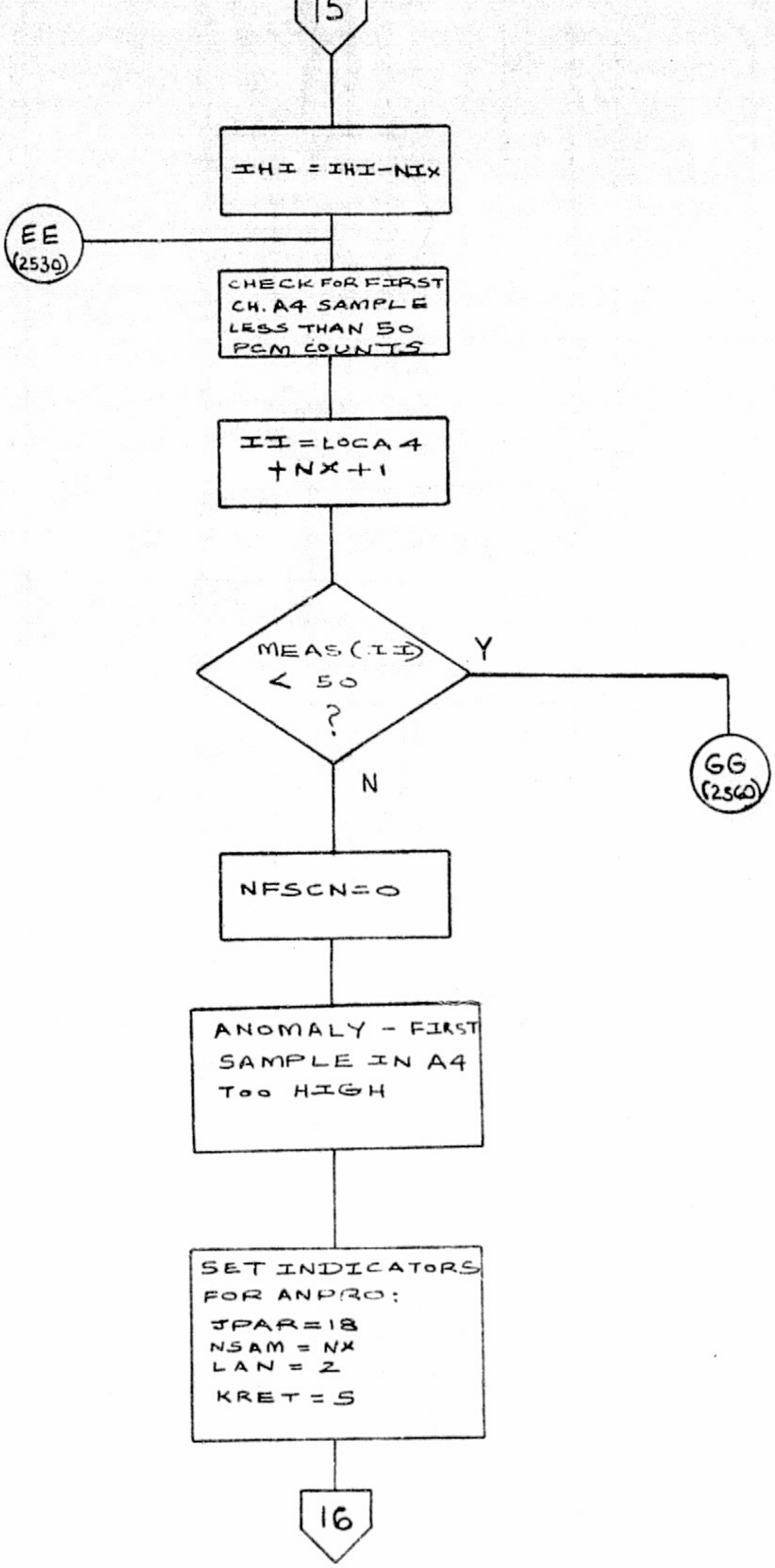

10
Z (2223)
LAN = 3
CC (2230)
AA (2225)
LAN = 5
CC (2230)
JPAR = 16
NSAM = NX
KRET = 5
SEG
CALL IN
ANPRO.LLL
TO OUTPUT
AN ANOMALY
MESSAGE
BB (2500)
S (2300)
PREVIOUS
SCAN WAS
INVALID—
ZERO BIAS
HISTOGRAM
ACCUMULATORS
11

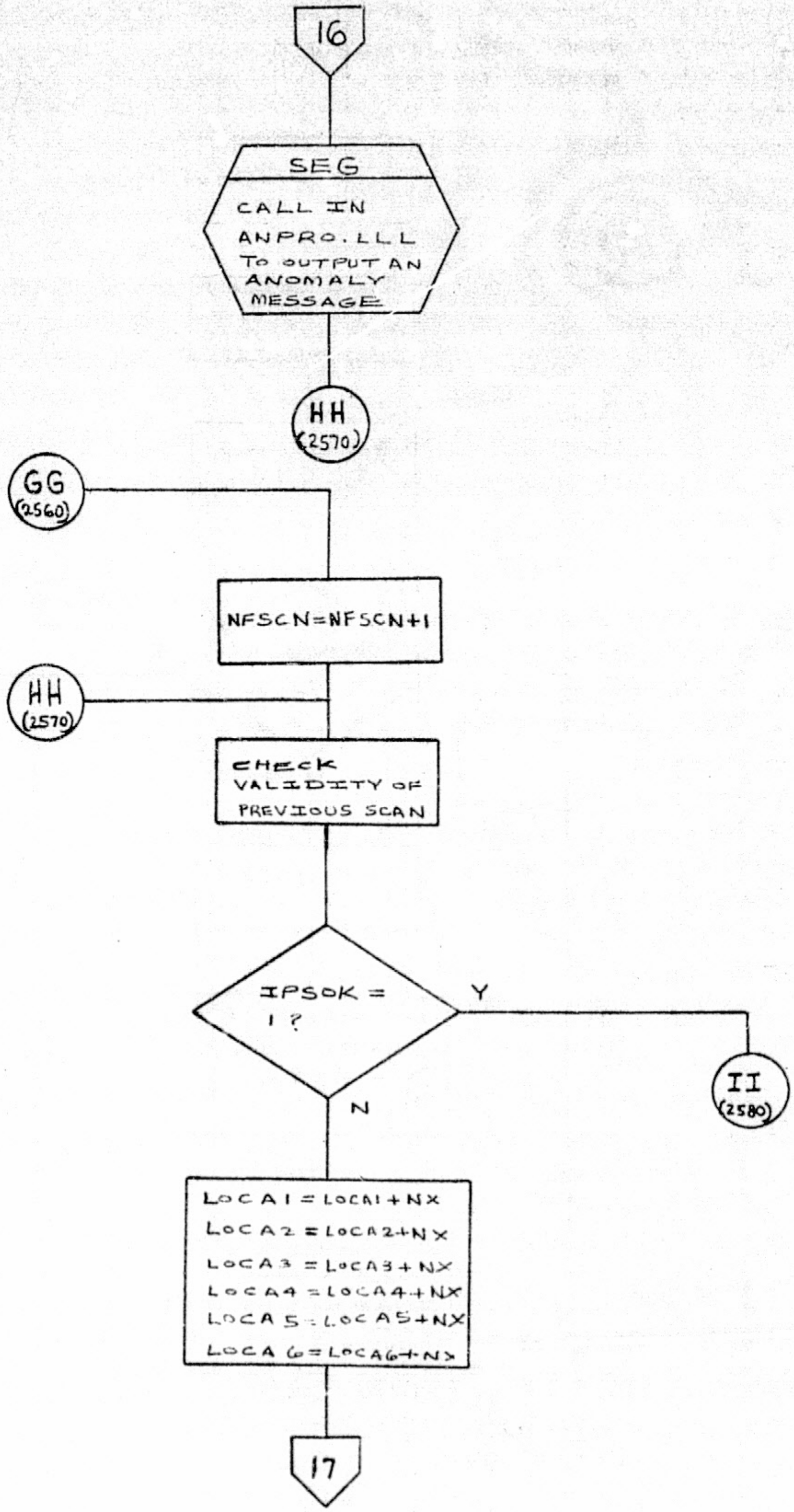

11
ZERO
CLEAR IBHIS1 ARRAY
ZERO
CLEAR IBHIS2 ARRAY
ZERO
CLEAR IBHIS3 ARRAY
ZERO
CLEAR IBHIS5 ARRAY
ZERO
CLEAR IBHIS6 ARRAY
ISSN=0
NSAMPL=1
12

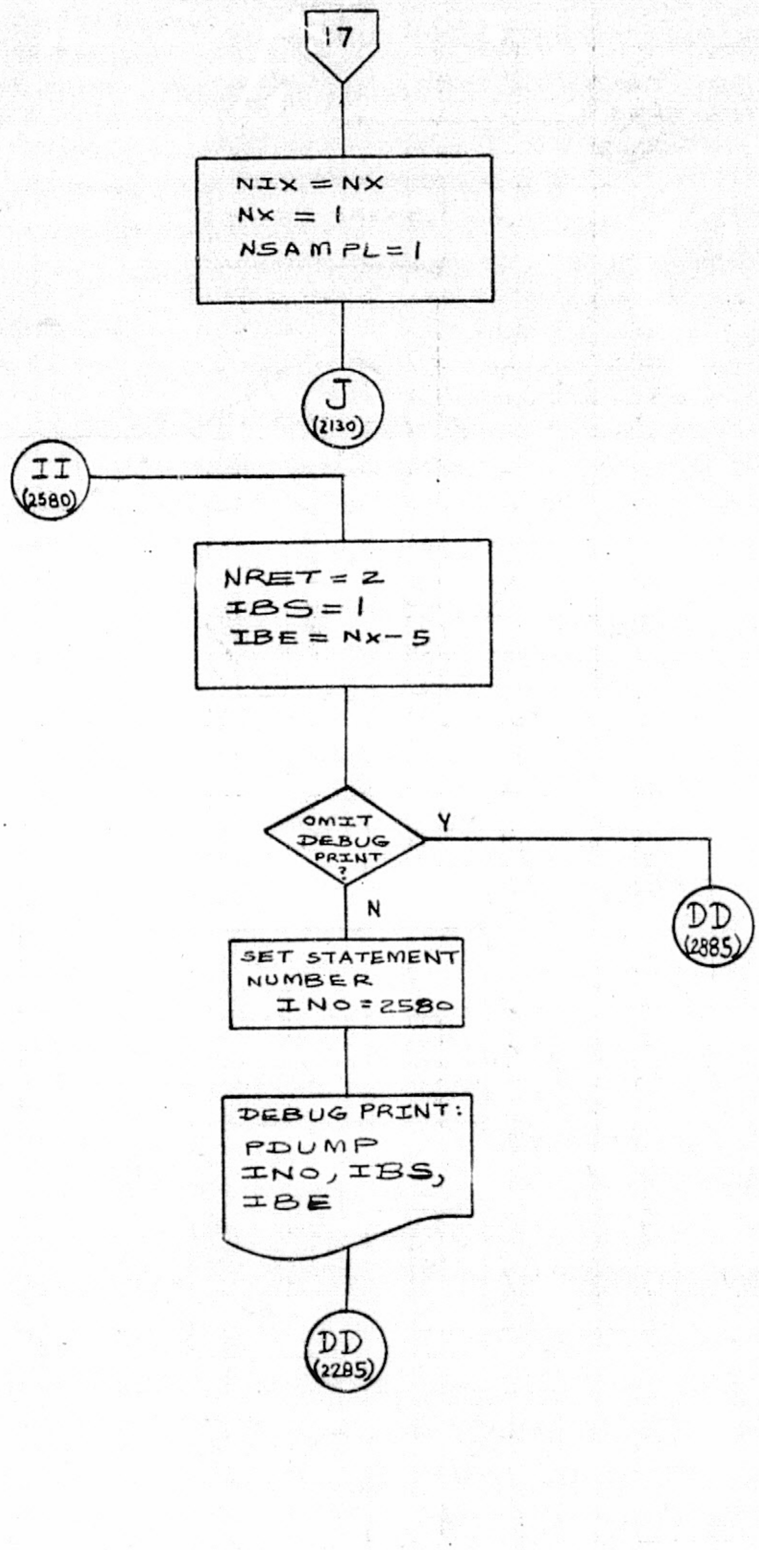

12
ANOMALY — BAD SYNC PULSE ON CHANNEL A6
SET INDICATORS FOR ANPRO:
NFSCN=1
JPAR=16
NSAM=NX
LAN =6
KRET=5
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
N (2170)
V (2400)
END OF MEAS ARRAY DATA - CHECK DATA FOR HISTOGRAM VALUES
ISTA=1
NRET=3
IBS= NSAMPL
IBE = NSAMPL+NX-5
13

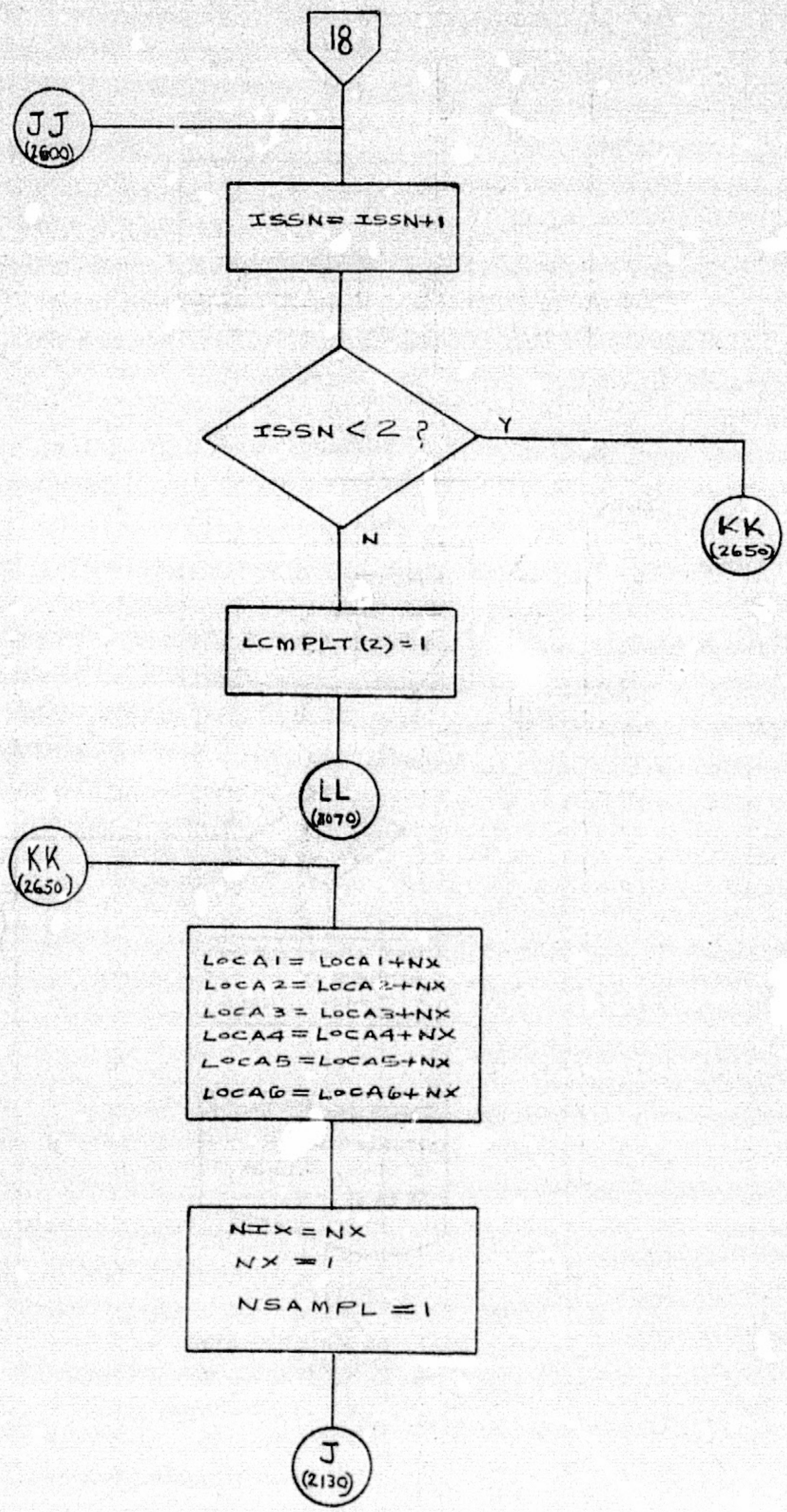

13
OMIT DEBUG PRINT ?
Y
N
DD (2885)
SET STATEMENT NUMBER
INO=2400
DEBUG PRINT:
PDUMP
INO, IBS,
IBE
DD (2885)
BB (2500)
NFSCN=0?
Y
N
EE (2530)
CHECK FOR CORRECT NO. OF SAMPLES IN SCAN
(NSAMPL+NX)
>ISMAX
Y
FF (2510)
14

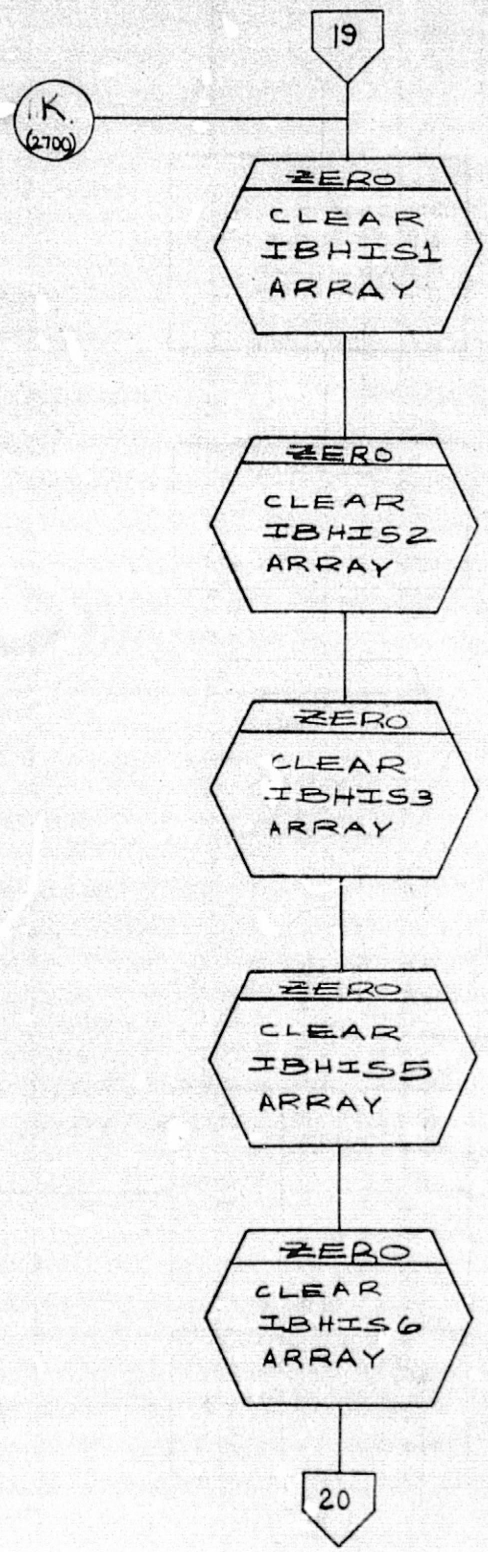

14
(NSAMPL+NX) < ISMIN ?
Y
N
FF (2510)
IPSOK = 1
EE (2530)
FF (2510)
ANOMALY - SCAN TOO LONG OR TOO SHORT
SET INDICATORS FOR ANPRO:
JPAR = 18
NSAM = NX
IPSOK = 0
NFSCN = 0
LAN=NSAMPL+NX
KRET=5
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
15

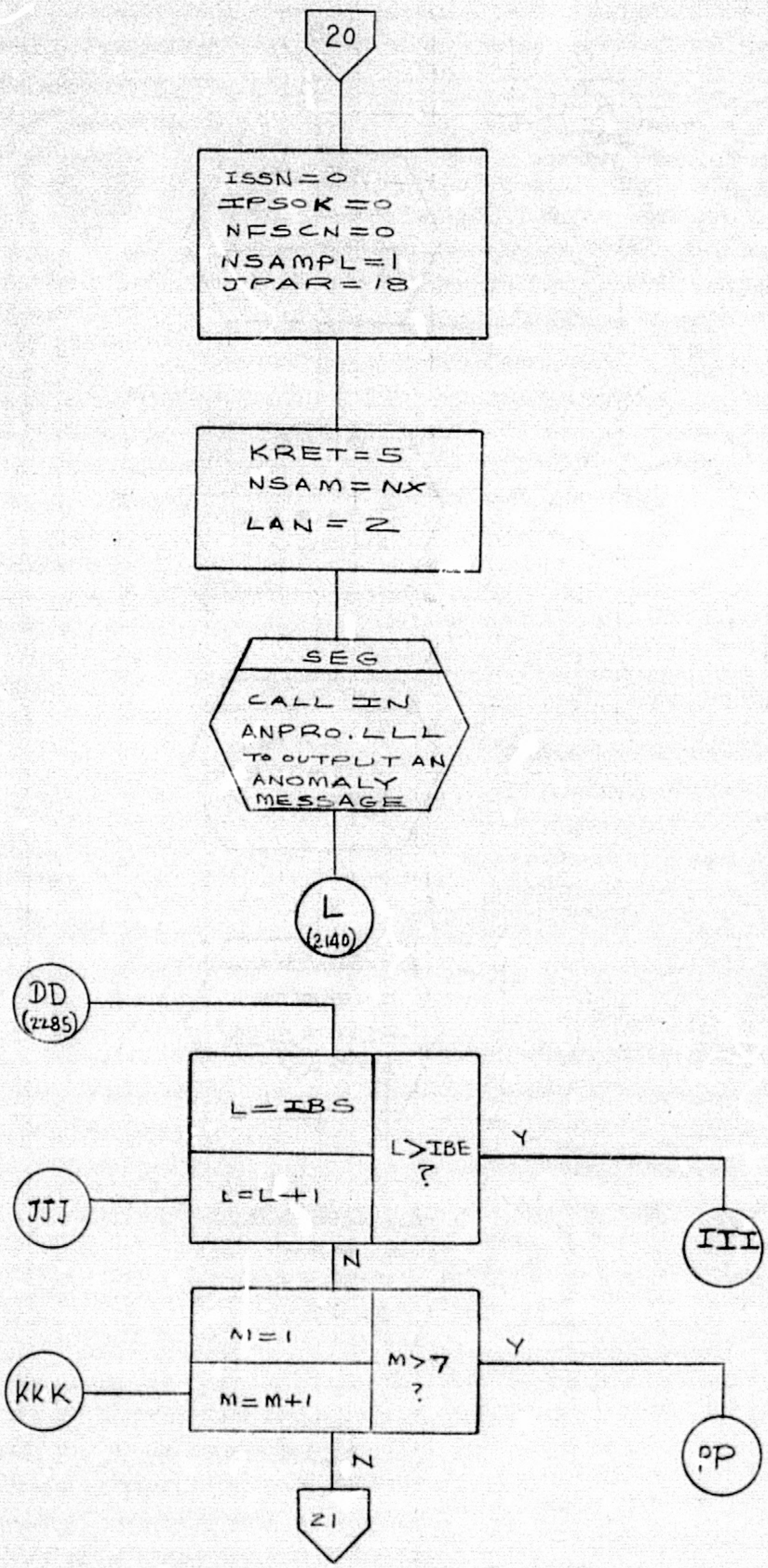

15
IHI = IHI-NIX
EE
(2530)
CHECK FOR FIRST CH. A4 SAMPLE LESS THAN 50 PCM COUNTS
II = LOCA4 +NX+1
MEAS(II) < 50 ?
Y
GG
(2560)
N
NFSCN=0
ANOMALY - FIRST SAMPLE IN A4 TOO HIGH
SET INDICATORS FOR ANPRO:
JPAR=18
NSAM = NX
LAN = 2
KRET = 5
16

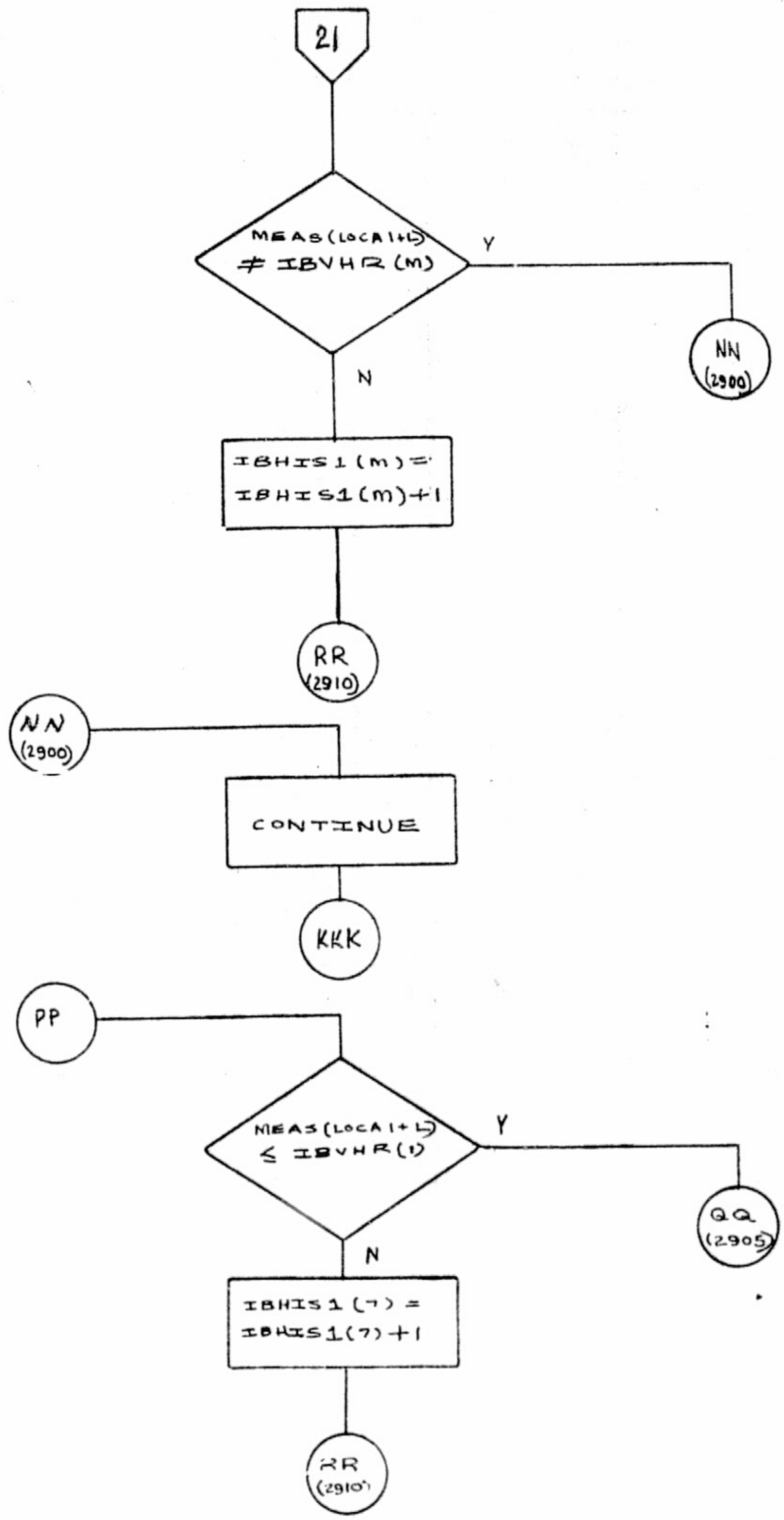

16
SEG
CALL IN
ANPRO.LLL
TO OUTPUT AN
ANOMALY
MESSAGE
HH
(2570)
GG
(2560)
NFSCN=NFSCN+1
HH
(2570)
CHECK
VALIDITY OF
PREVIOUS SCAN
IPSOK =
1 ?
Y
II
(2580)
N
LOCA1 = LOCA1 + NX
LOCA2 = LOCA2 + NX
LOCA3 = LOCA3 + NX
LOCA4 = LOCA4 + NX
LOCA5 = LOCA5 + NX
LOCA6 = LOCA6 + NX
17

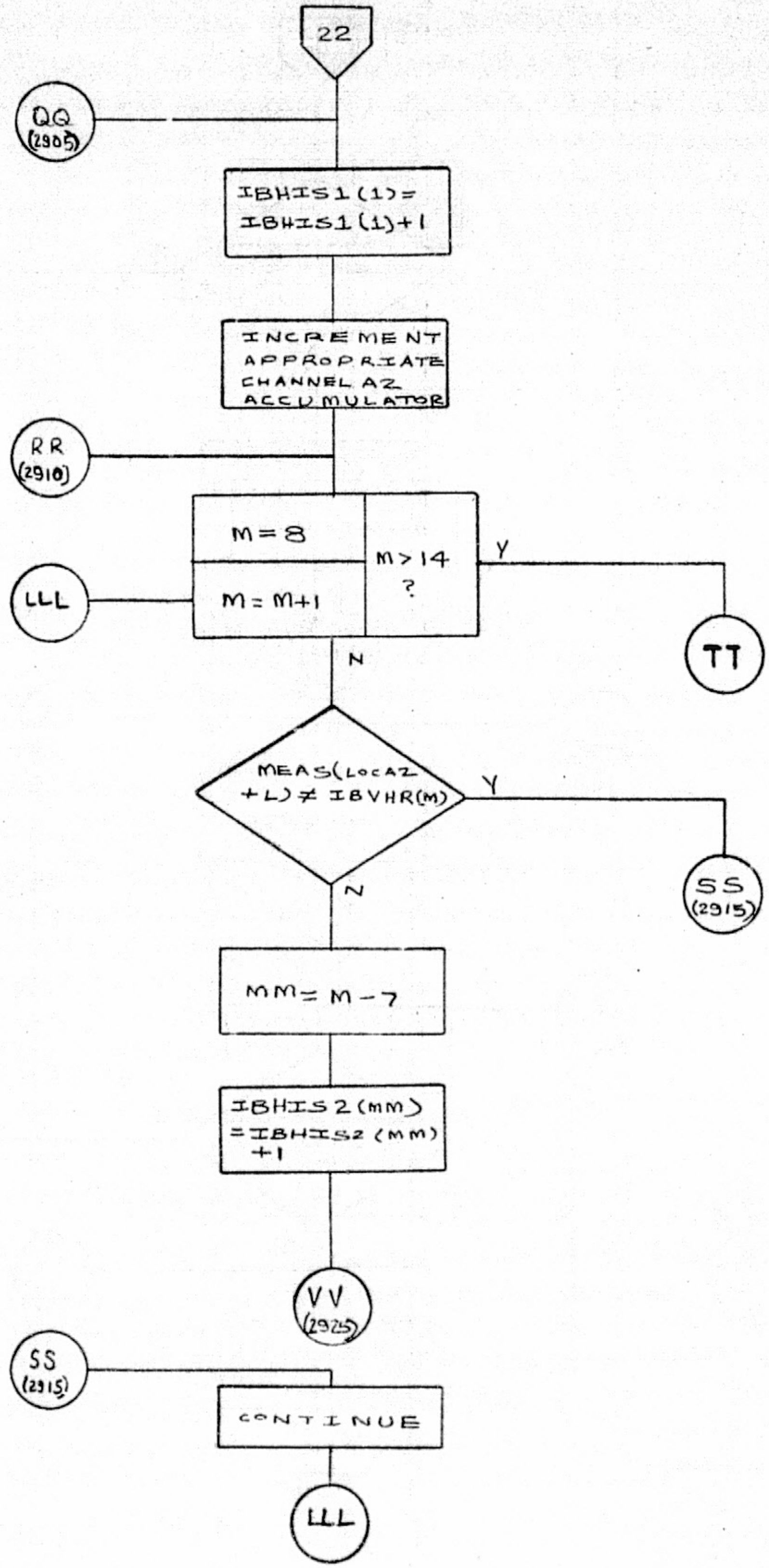

17
NIX = NX
NX = 1
NSAMPL = 1
J
(2130)
II
(2580)
NRET = 2
IBS = 1
IBE = NX - 5
OMIT DEBUG PRINT ?
Y
N
DD
(2885)
SET STATEMENT NUMBER
INO = 2580
DEBUG PRINT:
PDUMP
INO, IBS,
IBE
DD
(2285)

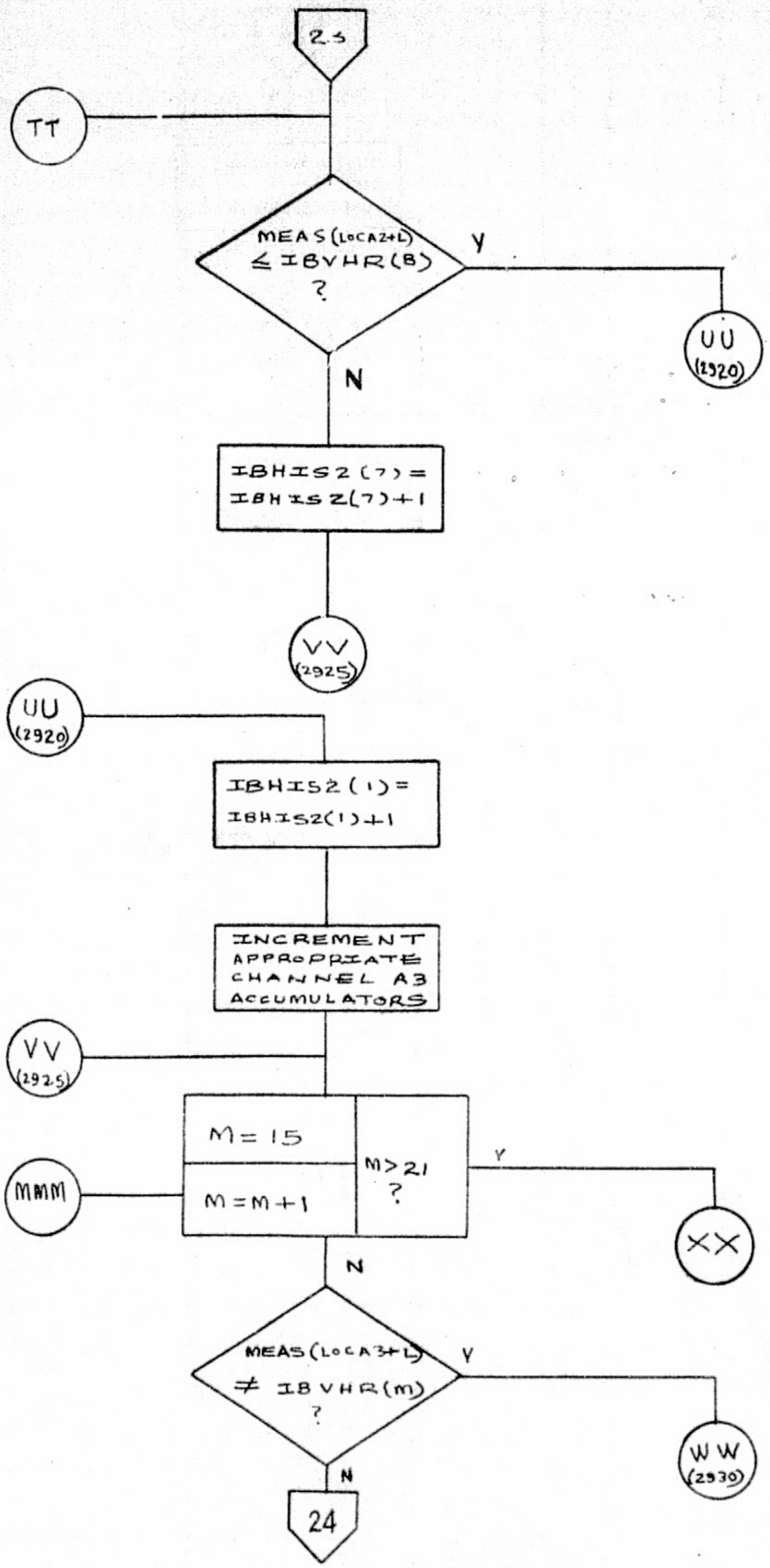

18
JJ (2600)
ISSN= ISSN+1
ISSN < 2 ?
Y
N
KK (2650)
LCMPLT (2) = 1
LL (3070)
KK (2650)
LOCA1 = LOCA1+NX
LOCA2 = LOCA2+NX
LOCA3 = LOCA3+NX
LOCA4 = LOCA4+NX
LOCA5 = LOCA5+NX
LOCA6 = LOCA6+NX
NIX = NX
NX = 1
NSAMPL = 1
J (2130)

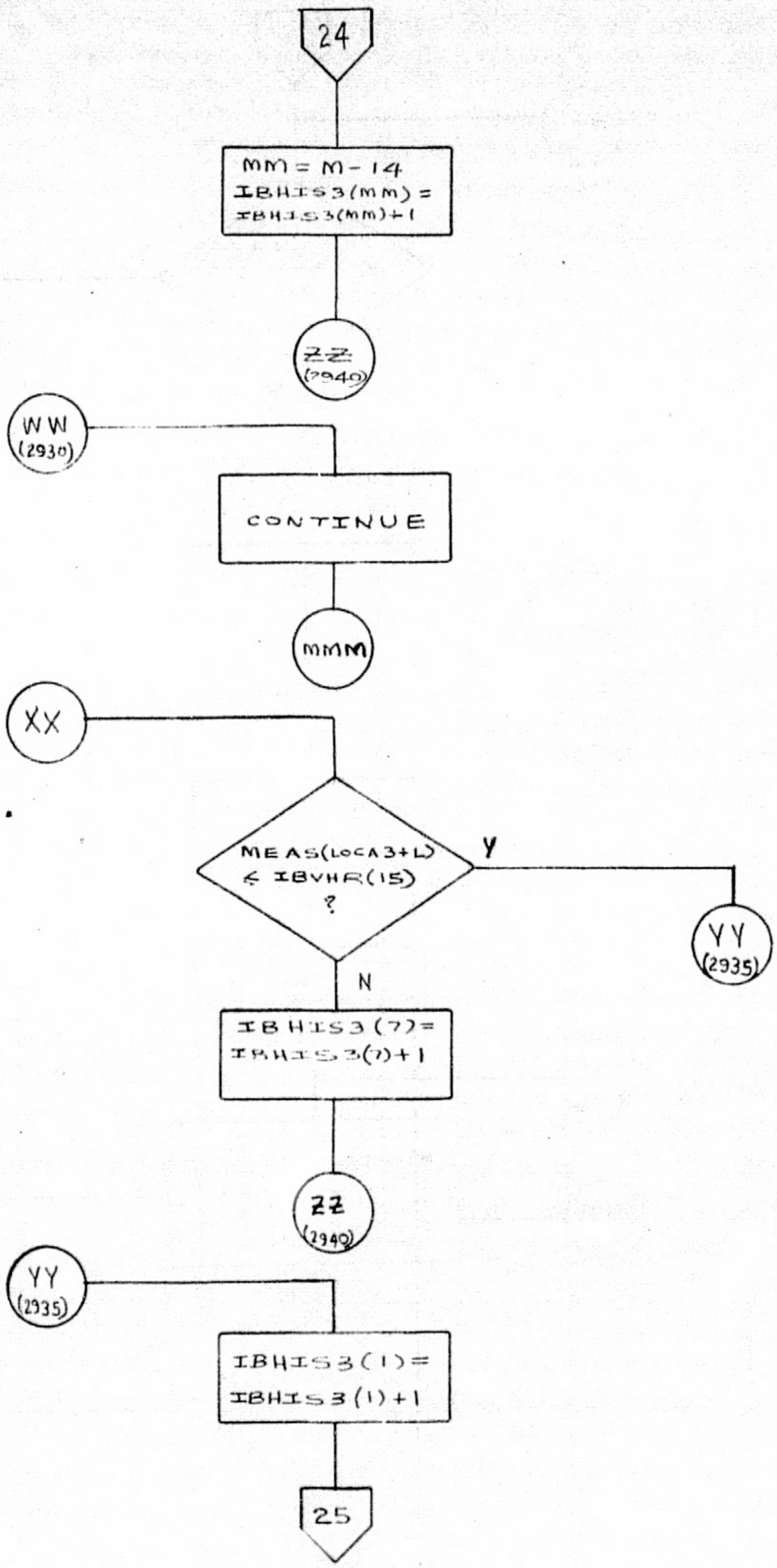

19
I.K.
(2700)
ZERO
CLEAR
IBHIS1
ARRAY
ZERO
CLEAR
IBHIS2
ARRAY
ZERO
CLEAR
IBHIS3
ARRAY
ZERO
CLEAR
IBHIS5
ARRAY
ZERO
CLEAR
IBHIS6
ARRAY
20

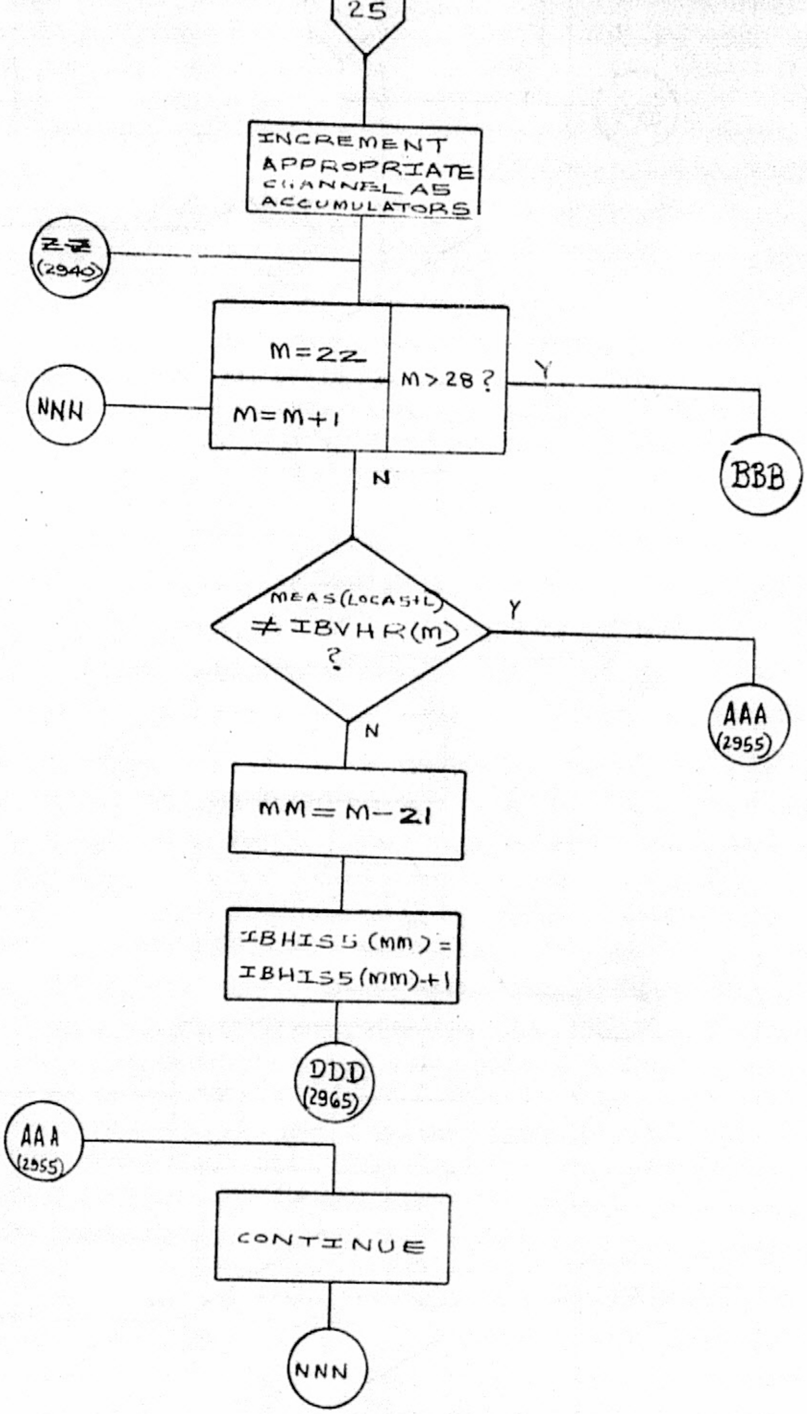

20
ISSN=0
IPSOK=0
NFSCN=0
NSAMPL=1
JPAR=18
KRET=5
NSAM=NX
LAN=2
SEG
CALL IN
ANPRO.LLL
TO OUTPUT AN
ANOMALY
MESSAGE
L
(2140)
DD
(2285)
L=IBS
L=L+1
L>IBE
?
Y
JJJ
III
N
M=1
M=M+1
M>7
?
Y
KKK
PP
N
21

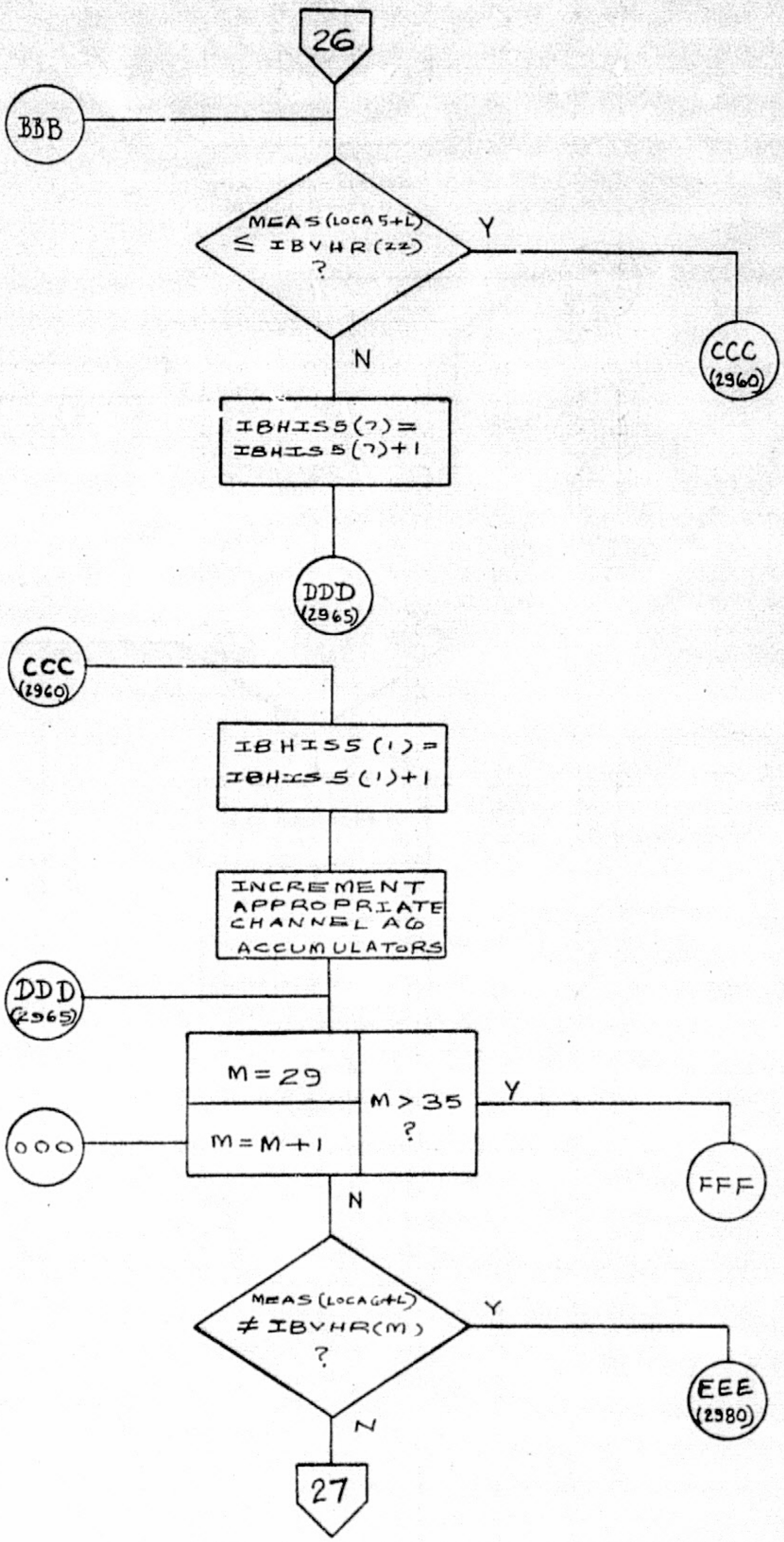

21
MEAS(LOCA1+L)
≠ IBVHR(M)
Y
NN
(2900)
N
IBHIS1(M)=
IBHIS1(M)+1
RR
(2910)
NN
(2900)
CONTINUE
KKK
PP
MEAS(LOCA1+L)
≤ IBVHR(1)
Y
QQ
(2905)
N
IBHIS1(7) =
IBHIS1(7)+1
RR
(2910)

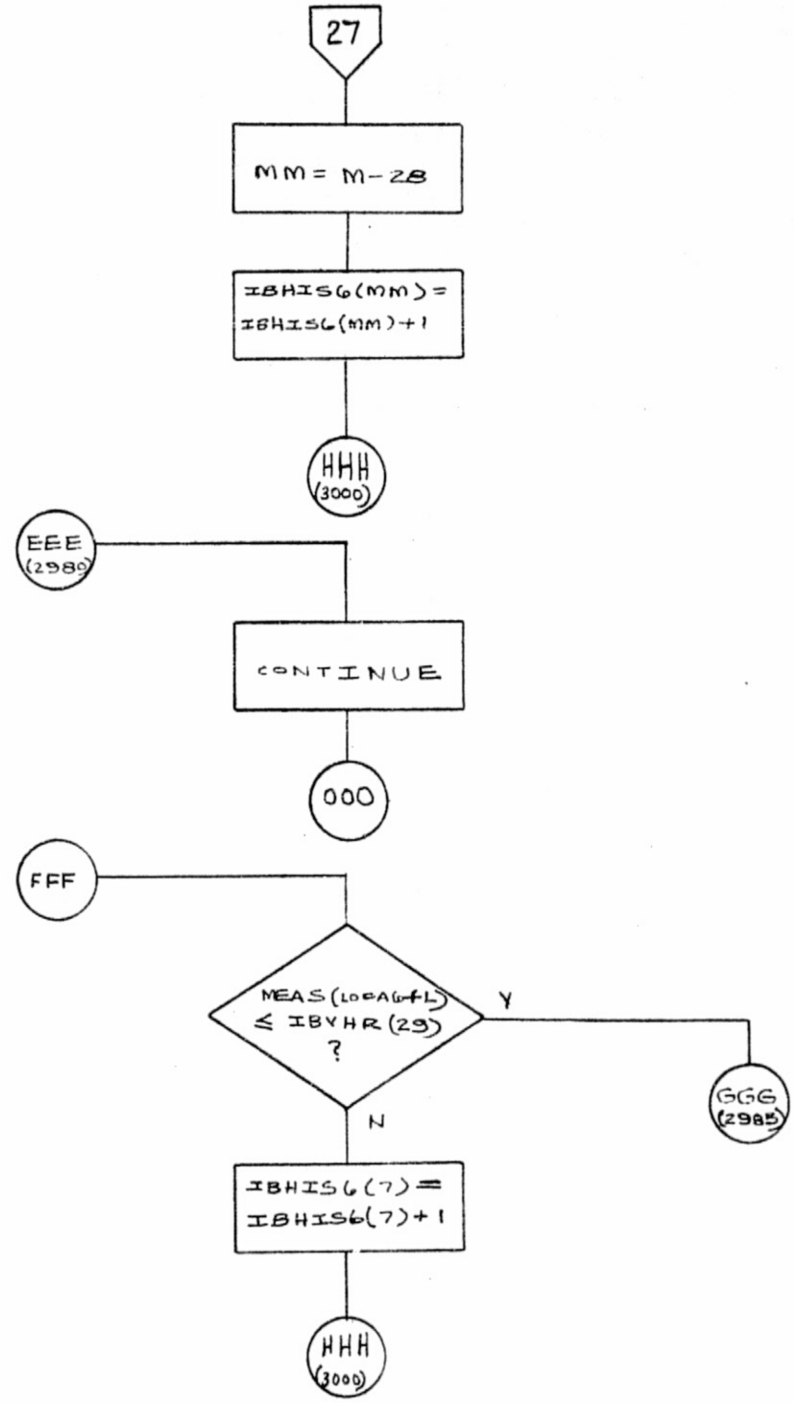

22
QQ
(2905)
IBHIS1(1)=
IBHIS1(1)+1
INCREMENT
APPROPRIATE
CHANNEL AZ
ACCUMULATOR
RR
(2910)
M=8
M=M+1
M>14
?
LLL
Y
TT
N
MEAS(LOCAZ
+L) ≠ IBVHR(M)
Y
SS
(2915)
N
MM= M-7
IBHIS2(MM)
=IBHIS2(MM)
+1
VV
(2925)
SS
(2915)
CONTINUE
LLL

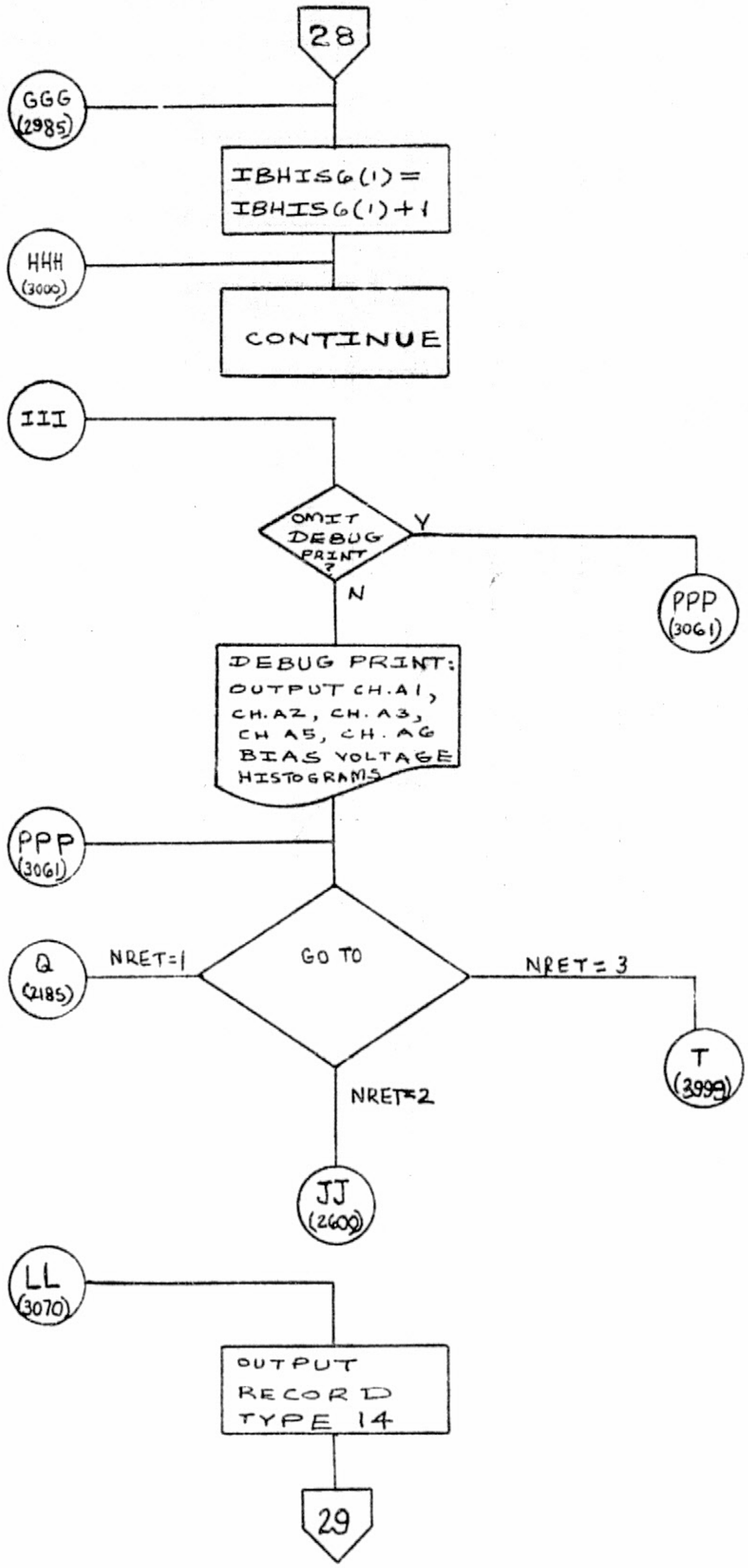

2.5
TT
MEAS (LOCA2+L) ≤ IBVHR(8) ?
Y
N
UU (2920)
IBHIS2(7) = IBHIS2(7)+1
VV (2925)
UU (2920)
IBHIS2(1) = IBHIS2(1)+1
INCREMENT APPROPRIATE CHANNEL A3 ACCUMULATORS
VV (2925)
M = 15
MMM
M = M+1
M > 21 ?
Y
N
XX
MEAS (LOCA3+L) ≠ IBVHR(m) ?
Y
N
WW (2930)
24


24
MM = M - 14
IBHIS3(MM) =
IBHIS3(MM)+1
ZZ
(2940)
WW
(2930)
CONTINUE
MMM
XX
MEAS(LOCA3+L)
≤ IBVHR(15)
?
Y
YY
(2935)
N
IBHIS3(7)=
IBHIS3(7)+1
ZZ
(2940)
YY
(2935)
IBHIS3(1)=
IBHIS3(1)+1
25

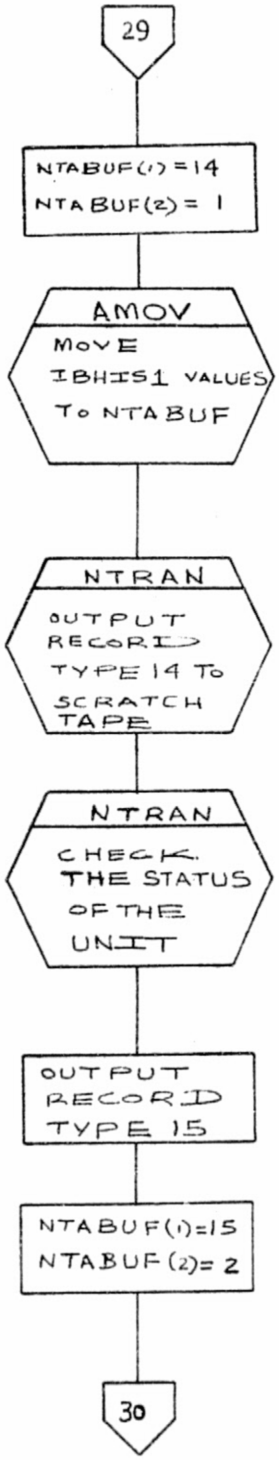

25
INCREMENT APPROPRIATE CHANNEL A5 ACCUMULATORS
ZZ (2940)
M=22
M=M+1
M>28 ?
Y
NNN
BBB
N
MEAS(LOCA5+L) ≠ IBVHR(M) ?
Y
AAA (2955)
N
MM=M−21
IBHIS5(MM)= IBHIS5(MM)+1
DDD (2965)
AAA (2955)
CONTINUE
NNN

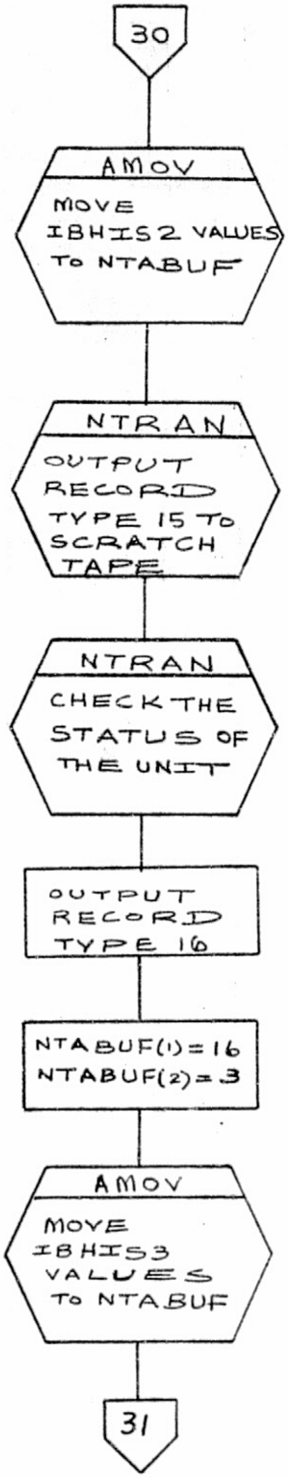

26
BBB
MEAS (LOCA5+L) ≤ IBVHR(22) ?
Y
N
CCC (2960)
IBHIS5(7)= IBHIS5(7)+1
DDD (2965)
CCC (2960)
IBHIS5(1)= IBHIS5(1)+1
INCREMENT APPROPRIATE CHANNEL A6 ACCUMULATORS
DDD (2965)
M=29
M=M+1
M>35 ?
Y
OOO
FFF
N
MEAS (LOCA6+L) ≠ IBVHR(M) ?
Y
EEE (2980)
N
27

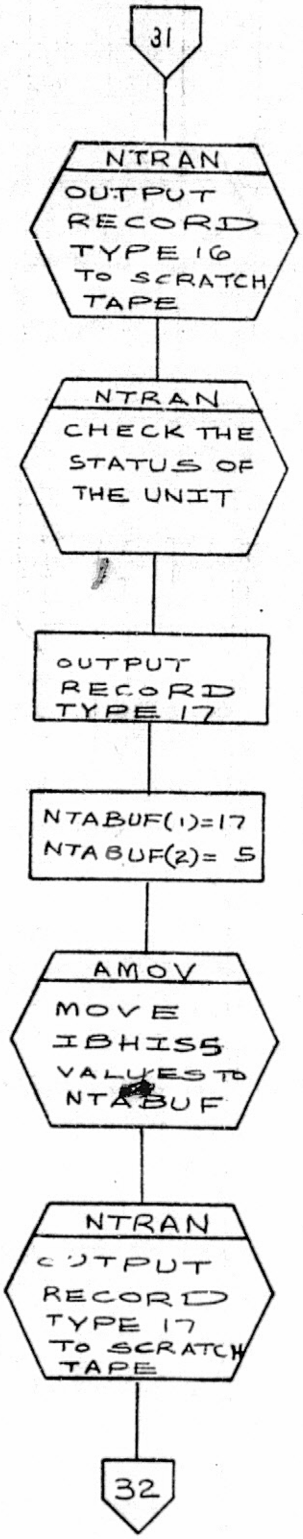

27
MM = M-28
IBHIS6(MM) =
IBHIS6(MM)+1
HHH
(3000)
EEE
(2980)
CONTINUE
000
FFF
MEAS(LOCAG+L)
≤ IBVHR(29)
?
Y
N
GGG
(2985)
IBHIS6(7) =
IBHIS6(7)+1
HHH
(3000)

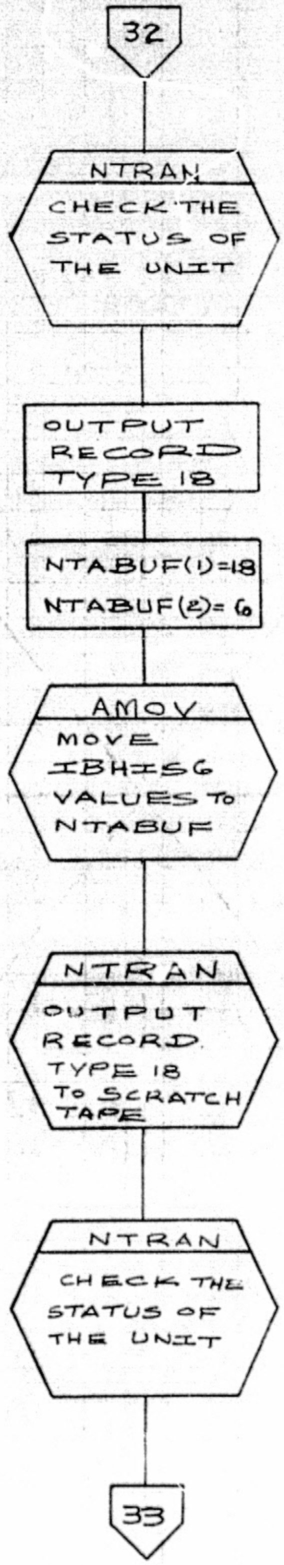


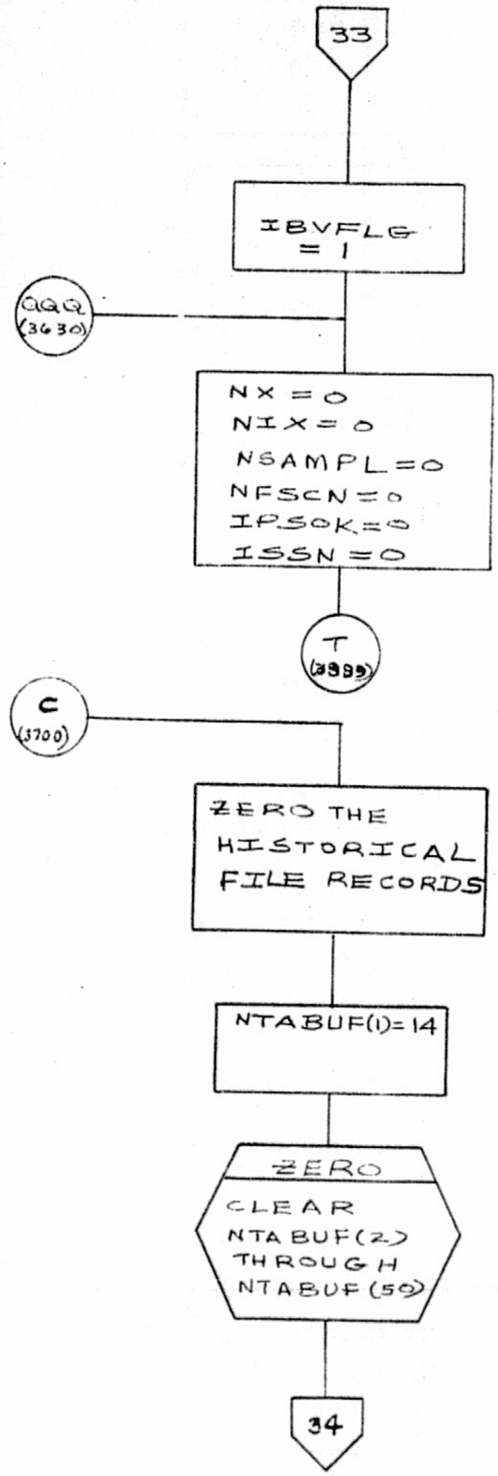
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

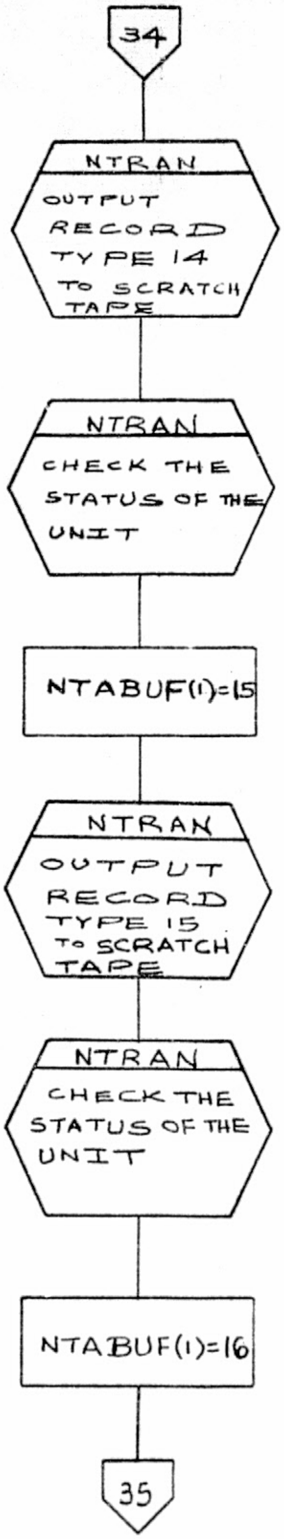

29
NTABUF(1)=14
NTABUF(2)= 1
AMOV
MOVE
IBHIS1 VALUES
TO NTABUF
NTRAN
OUTPUT
RECORD
TYPE 14 TO
SCRATCH
TAPE
NTRAN
CHECK
THE STATUS
OF THE
UNIT
OUTPUT
RECORD
TYPE 15
NTABUF(1)=15
NTABUF(2)= 2
30

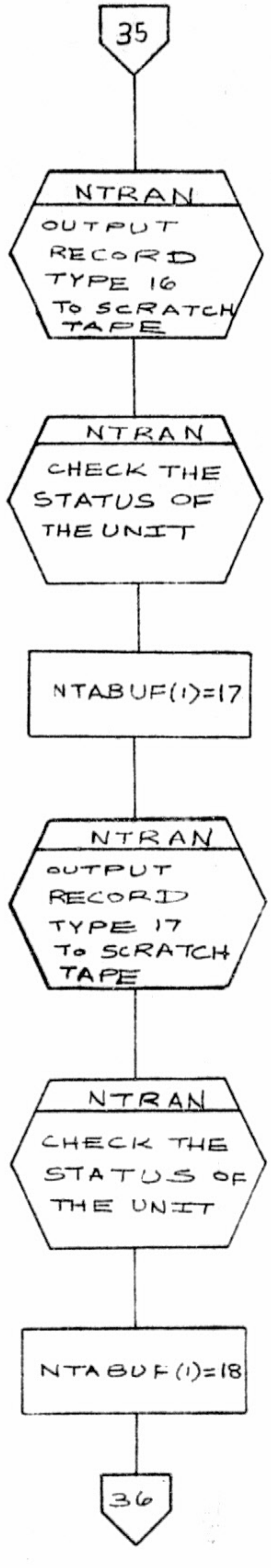

30
AMOV
MOVE IBHIS2 VALUES TO NTABUF
NTRAN
OUTPUT RECORD TYPE 15 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
OUTPUT RECORD TYPE 16
NTABUF(1)=16
NTABUF(2)=3
AMOV
MOVE IBHIS3 VALUES TO NTABUF
31

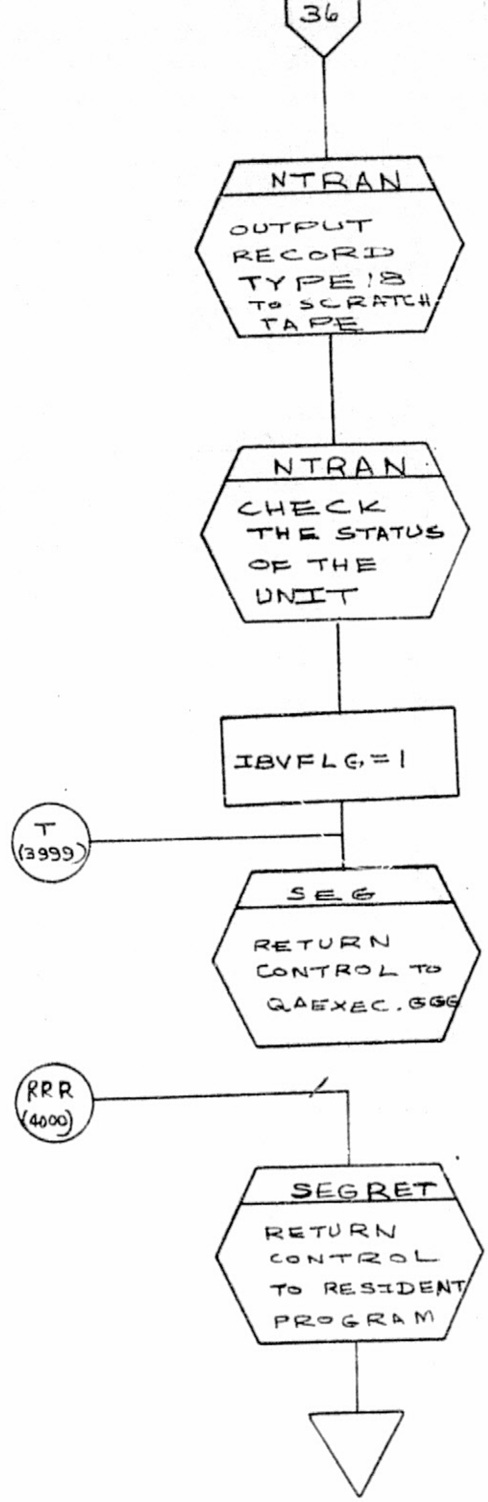

31
NTRAN
OUTPUT RECORD TYPE 16 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
OUTPUT RECORD TYPE 17
NTABUF(1)=17
NTABUF(2)= 5
AMOV
MOVE IBHIS5 VALUES TO NTABUF
NTRAN
OUTPUT RECORD TYPE 17 TO SCRATCH TAPE
32

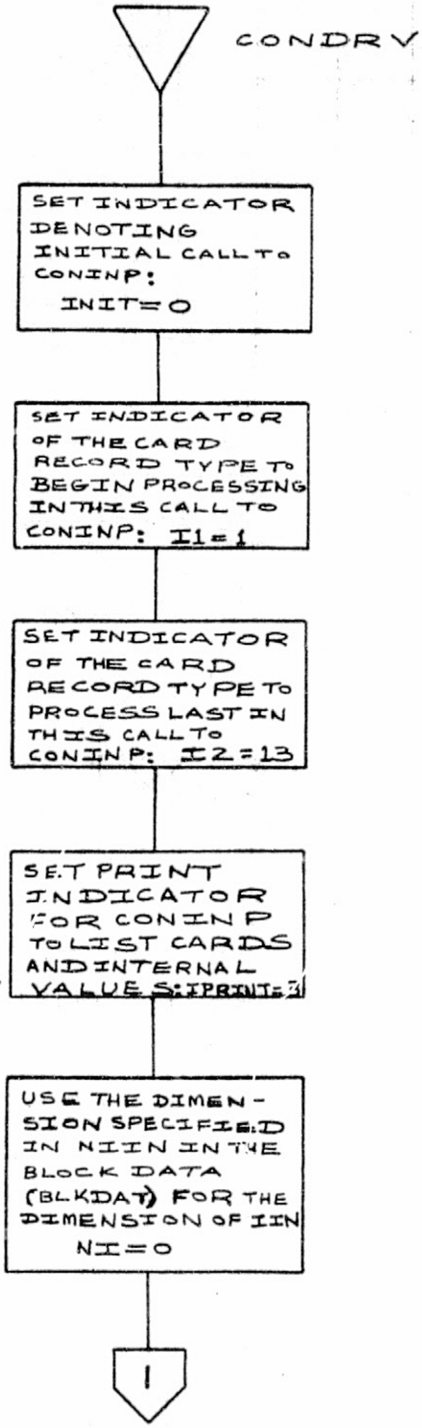

32
NTRAN
CHECK THE STATUS OF THE UNIT
OUTPUT RECORD TYPE 18
NTABUF(1)=18
NTABUF(2)=6
AMOV
MOVE IBHIS6 VALUES TO NTABUF
NTRAN
OUTPUT RECORD TYPE 18 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
33

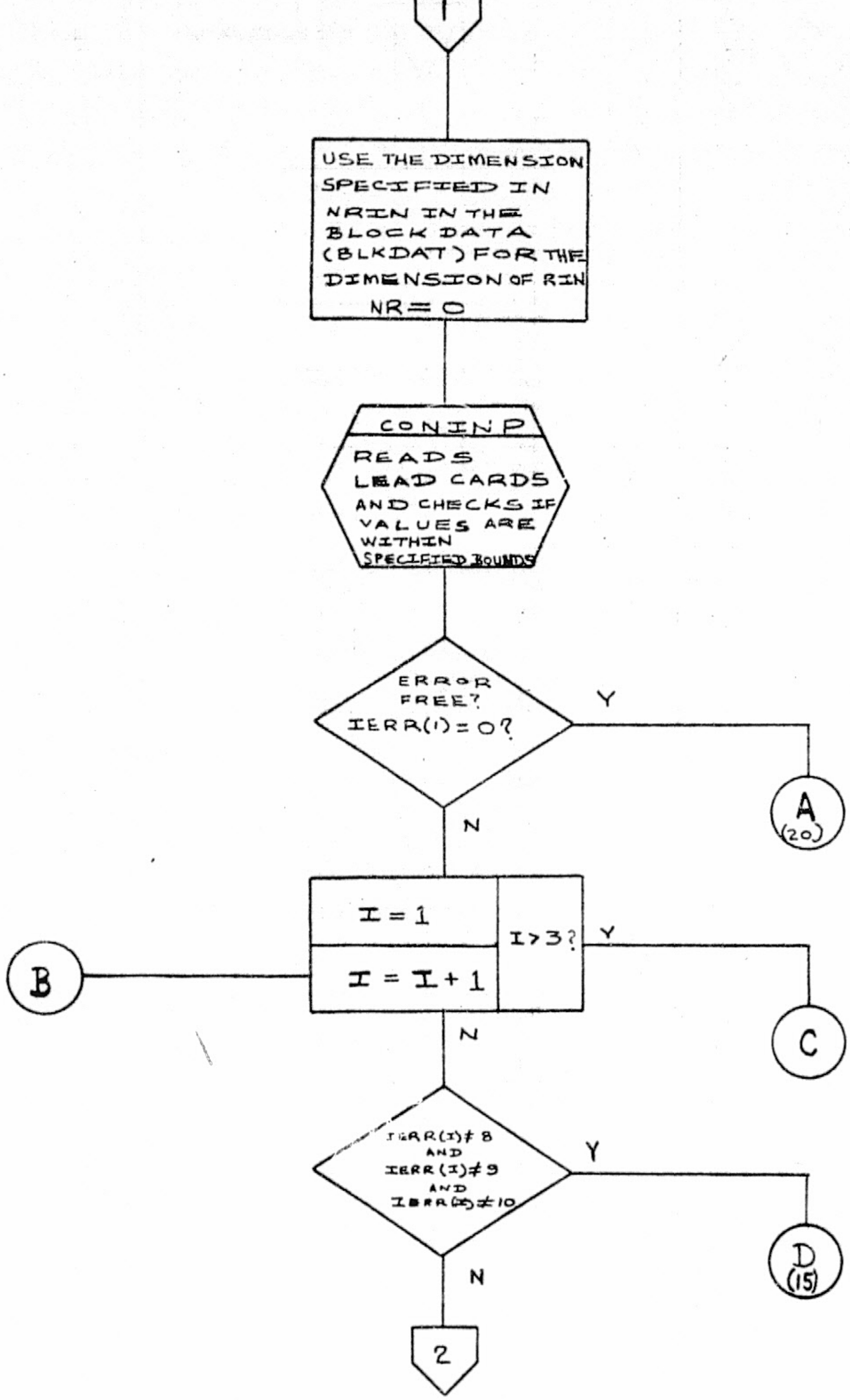

33
IBVFLG = 1
QQQ (3630)
NX = 0
NIX = 0
NSAMPL = 0
NFSCN = 0
IPSOK = 0
ISSN = 0
T (3999)
C (3700)
ZERO THE HISTORICAL FILE RECORDS
NTABUF(1)= 14
ZERO
CLEAR NTABUF(2) THROUGH NTABUF(50)
34

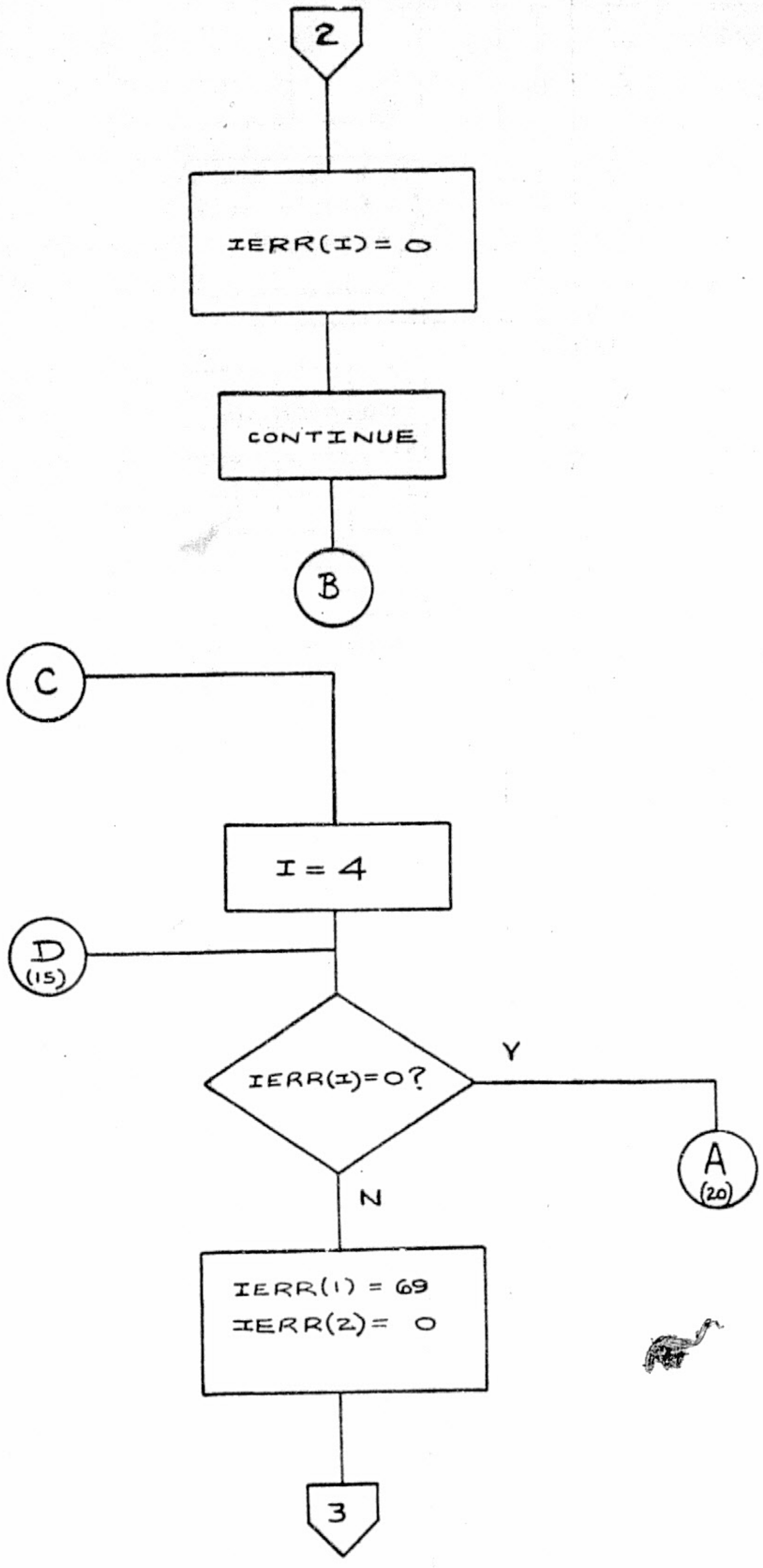

34
NTRAN
OUTPUT RECORD TYPE 14 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
NTABUF(1)=15
NTRAN
OUTPUT RECORD TYPE 15 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
NTABUF(1)=16
35

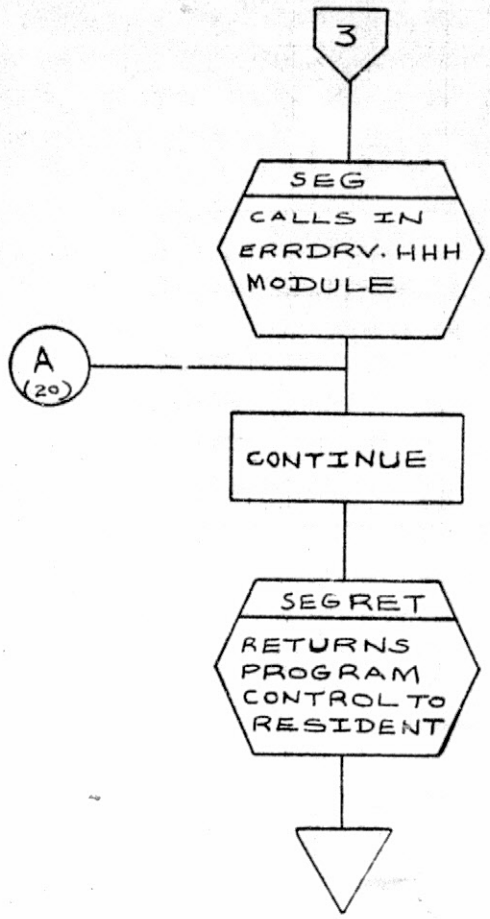

35
NTRAN
OUTPUT RECORD TYPE 16 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
NTABUF(1)=17
NTRAN
OUTPUT RECORD TYPE 17 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
NTABUF(1)=18
36

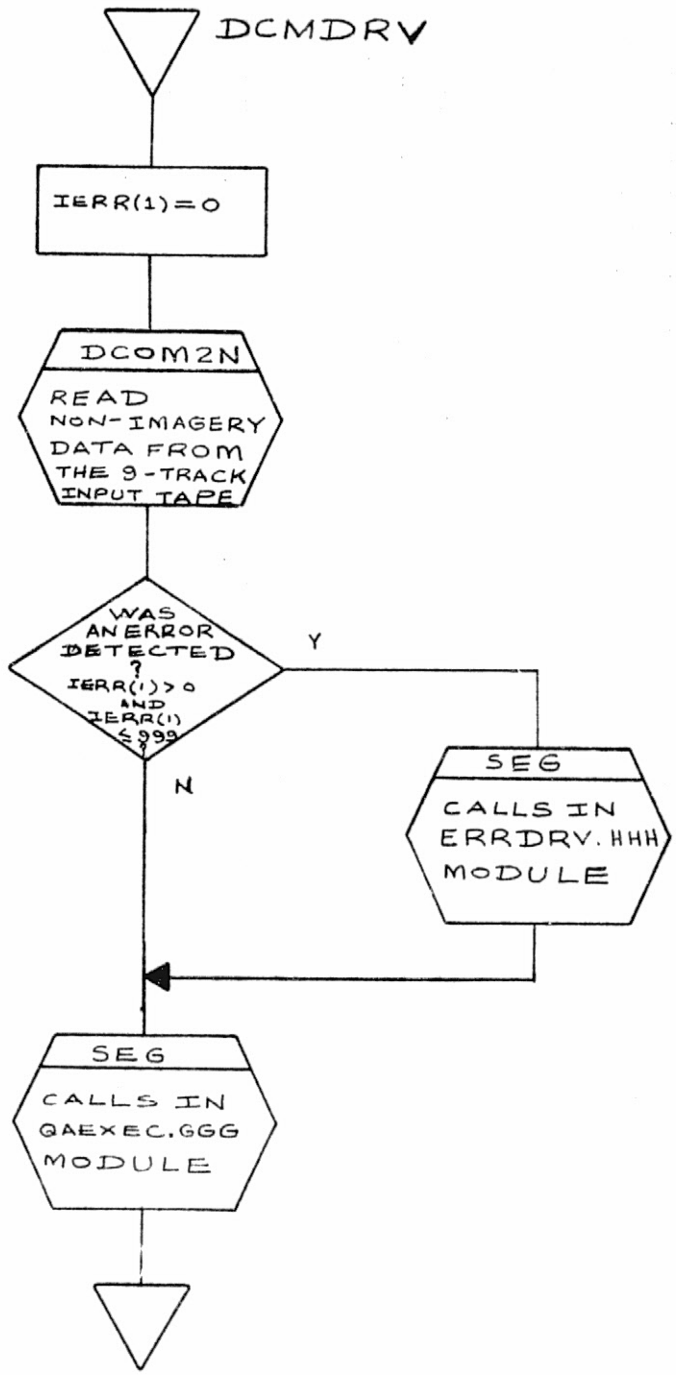

36
NTRAN
OUTPUT RECORD TYPE 18 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
IBVFLG=1
T (3999)
SEG
RETURN CONTROL TO QAEXEC.GGG
RRR (4000)
SEGRET
RETURN CONTROL TO RESIDENT PROGRAM

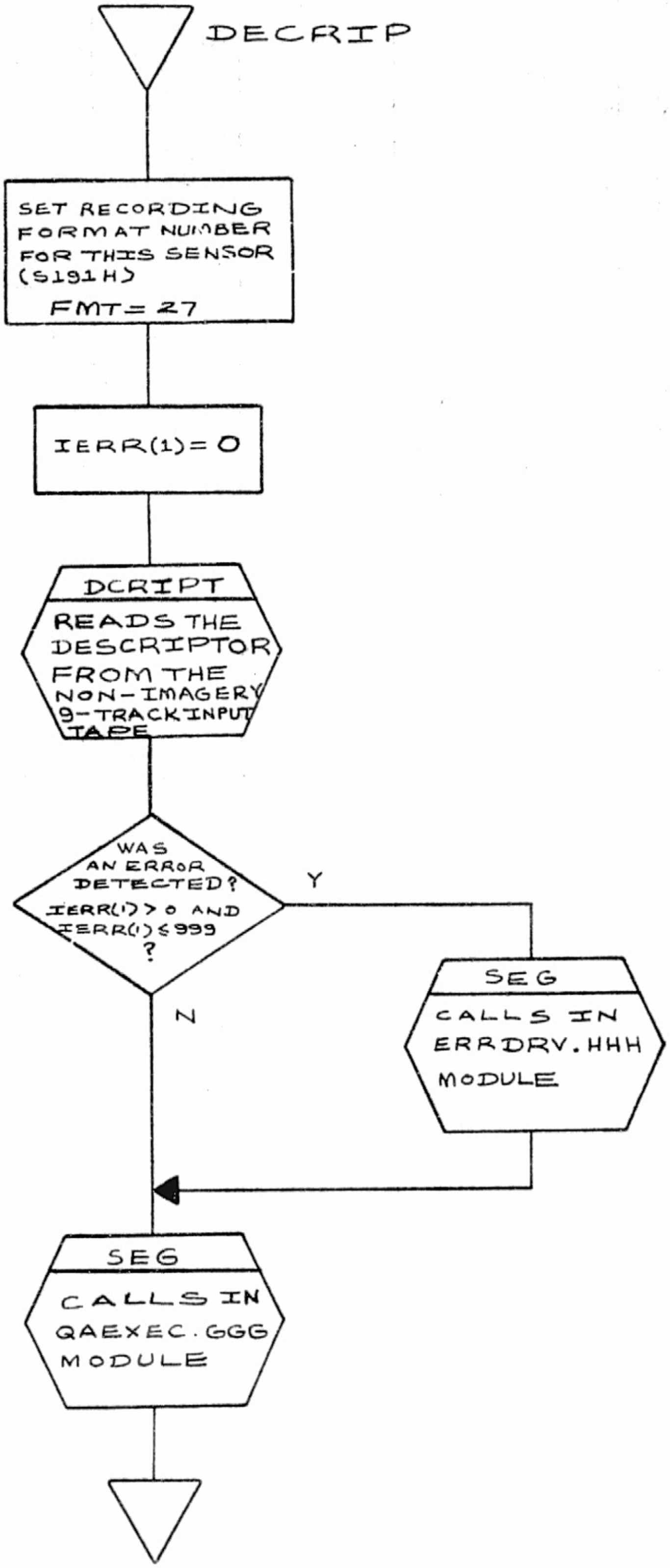

CONDRV
SET INDICATOR DENOTING INITIAL CALL TO CONINP: INIT=0
SET INDICATOR OF THE CARD RECORD TYPE TO BEGIN PROCESSING IN THIS CALL TO CONINP: I1=1
SET INDICATOR OF THE CARD RECORD TYPE TO PROCESS LAST IN THIS CALL TO CONINP: I2=13
SET PRINT INDICATOR FOR CONINP TO LIST CARDS AND INTERNAL VALUES: IPRINT=3
USE THE DIMENSION SPECIFIED IN NIIN IN THE BLOCK DATA (BLKDAT) FOR THE DIMENSION OF IIN NI=0
1

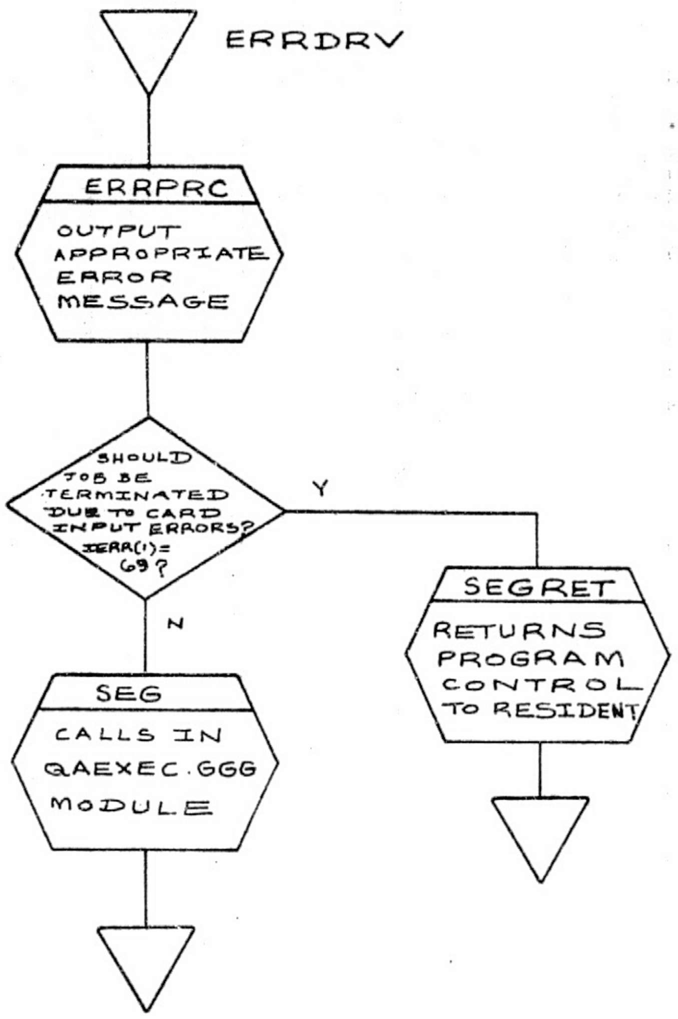

1
USE THE DIMENSION SPECIFIED IN NRIN IN THE BLOCK DATA (BLKDAT) FOR THE DIMENSION OF RIN
NR = 0
CONINP
READS LEAD CARDS AND CHECKS IF VALUES ARE WITHIN SPECIFIED BOUNDS
ERROR FREE? IERR(1) = 0?
Y
A (20)
N
I = 1
I = I + 1
I>3?
Y
B
C
N
IERR(I)≠8 AND IERR(I)≠9 AND IERR(I)≠10
Y
D (15)
N
2

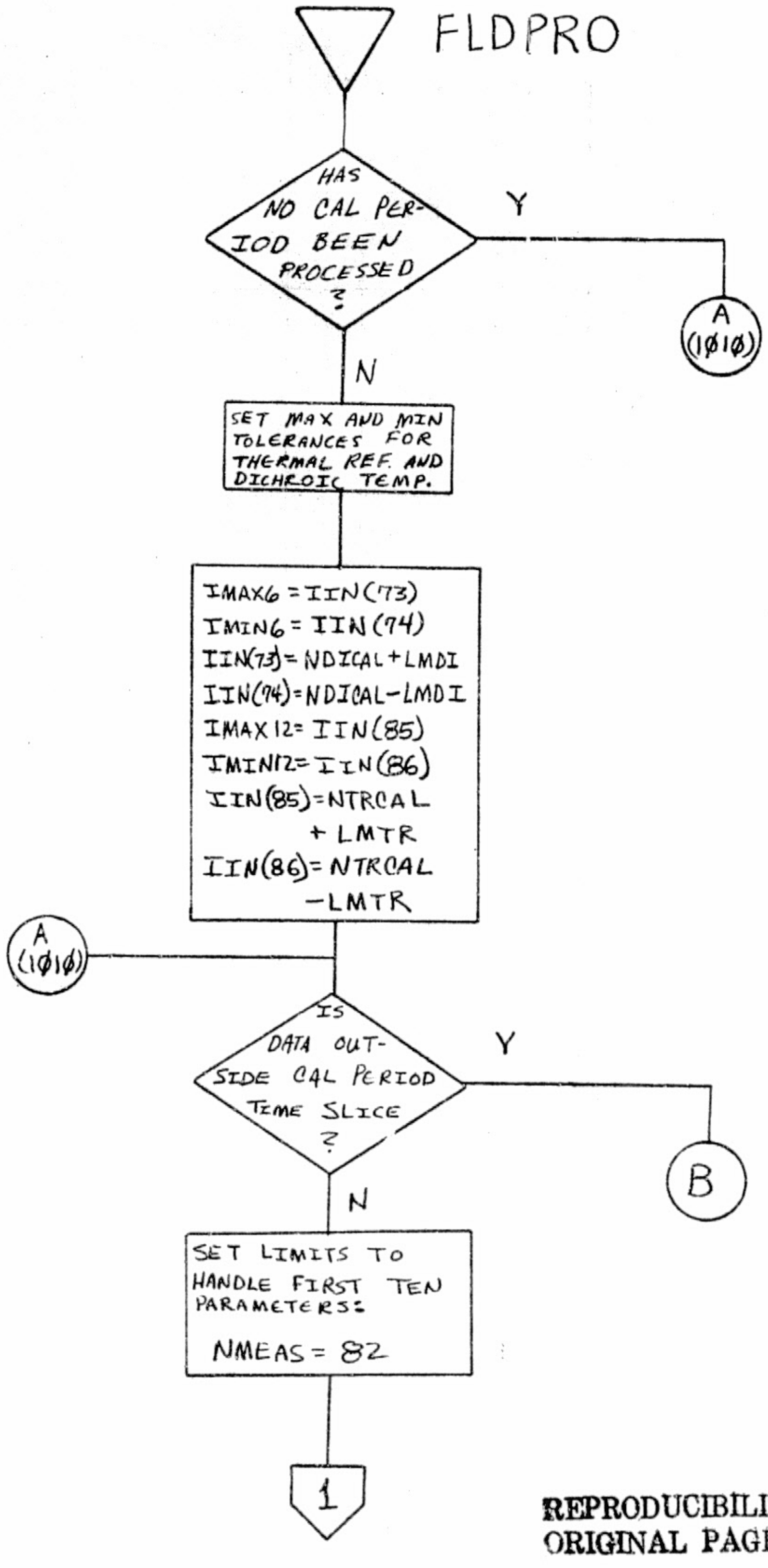

2
IERR(I) = 0
CONTINUE
B
C
I = 4
D
(15)
IERR(I)=0?
Y
A
(20)
N
IERR(1) = 69
IERR(2) = 0
3

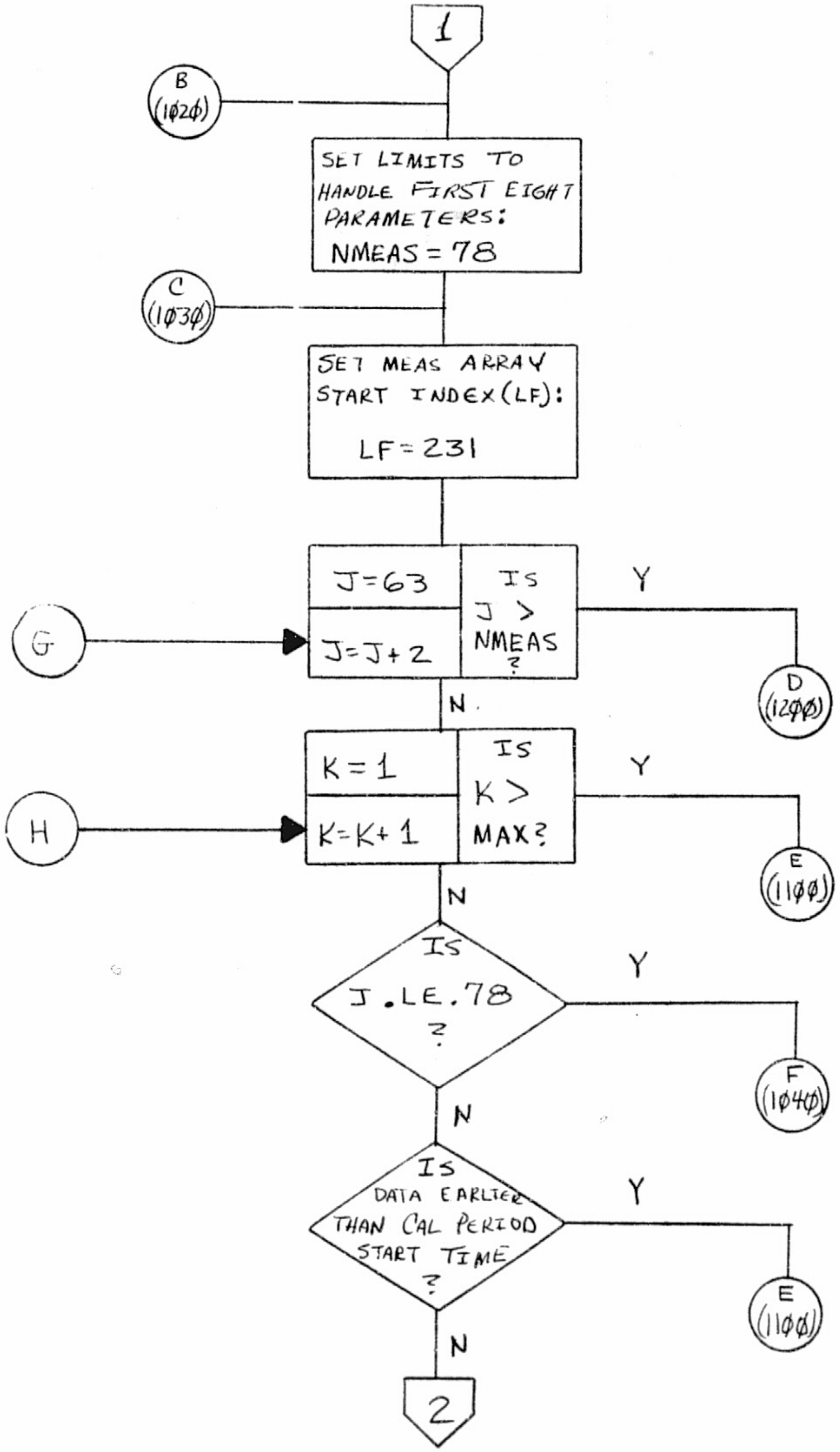

3
SEG
CALLS IN ERRDRV.HHH MODULE
A
(20)
CONTINUE
SEGRET
RETURNS PROGRAM CONTROL TO RESIDENT

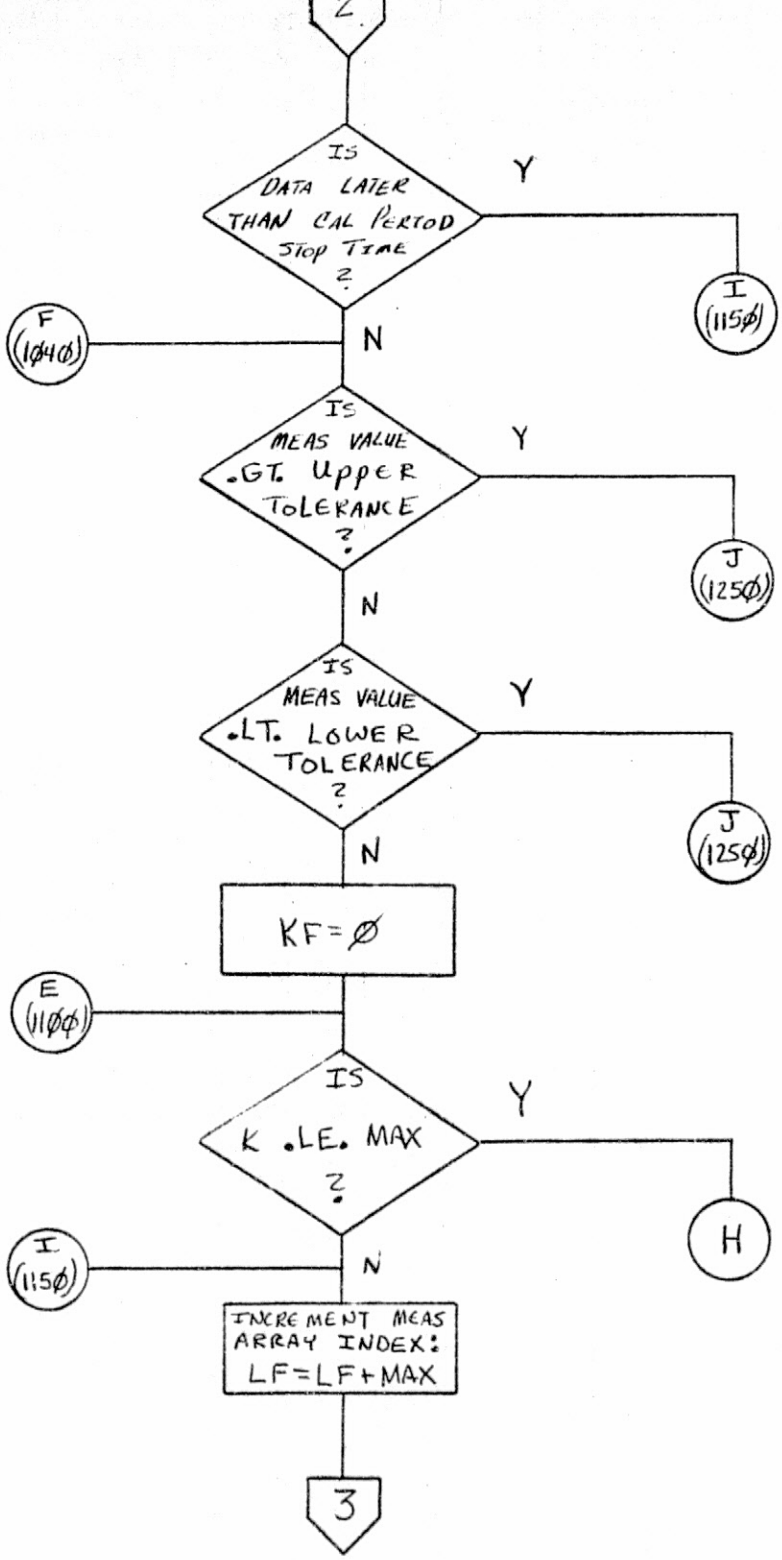

DCMDRV
IERR(1)=0
DCOM2N
READ NON-IMAGERY DATA FROM THE 9-TRACK INPUT TAPE
WAS AN ERROR DETECTED ? IERR(1)>0 AND IERR(1) ≤999 ?
Y
N
SEG
CALLS IN ERRDRV.HHH MODULE
SEG
CALLS IN QAEXEC.GGG MODULE

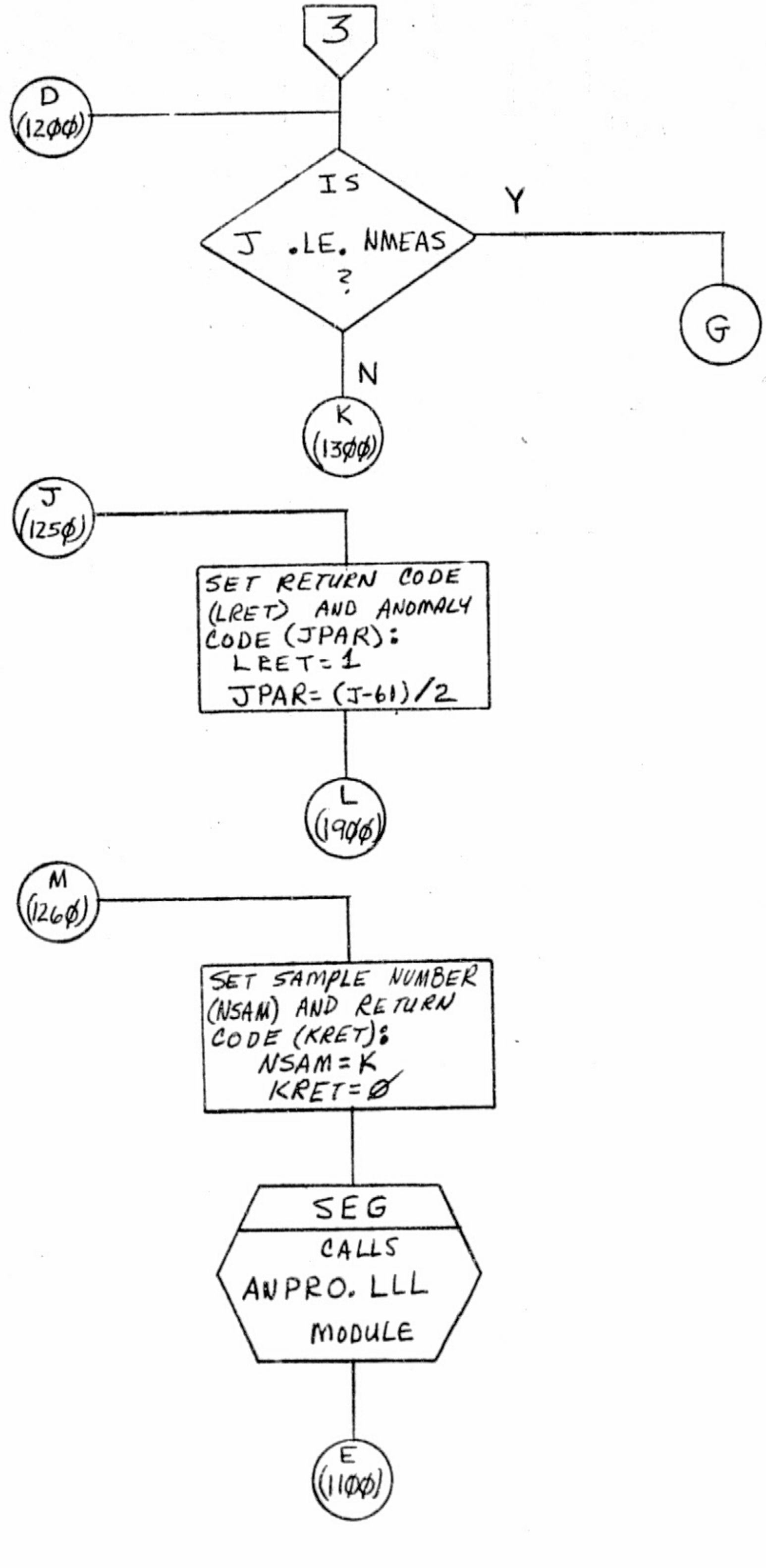

DECRIP
SET RECORDING FORMAT NUMBER FOR THIS SENSOR (S191H)
FMT= 27
IERR(1)= 0
DCRIPT
READS THE DESCRIPTOR FROM THE NON-IMAGERY 9-TRACK INPUT TAPE
WAS AN ERROR DETECTED? IERR(1) > 0 AND IERR(1) ≤ 999 ?
Y
N
SEG
CALLS IN ERRDRV.HHH MODULE
SEG
CALLS IN QAEXEC.GGG MODULE

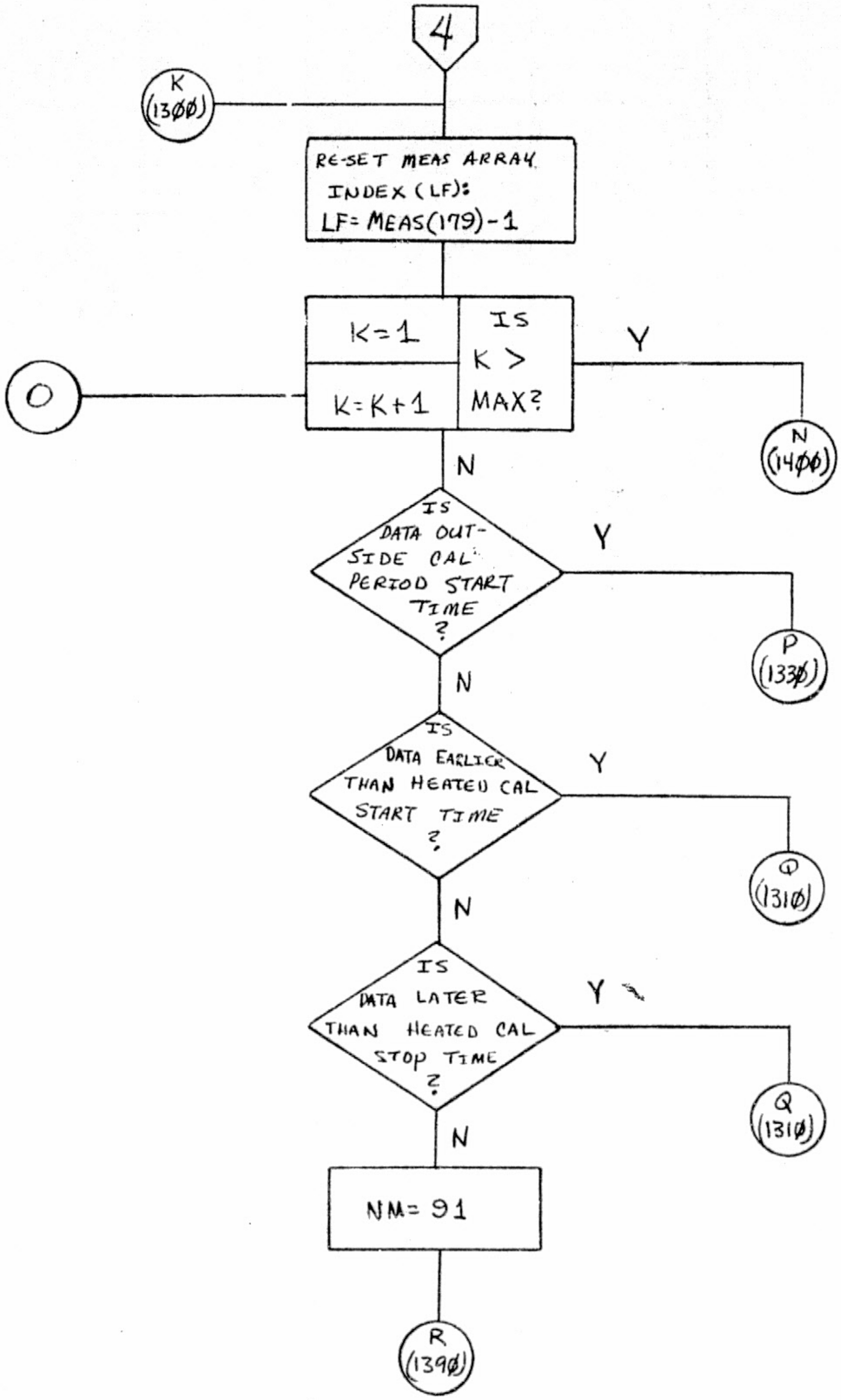

ERRDRV
ERRPRC
OUTPUT APPROPRIATE ERROR MESSAGE
SHOULD JOB BE TERMINATED DUE TO CARD INPUT ERRORS? IERR(1)= 69?
Y
N
SEGRET
RETURNS PROGRAM CONTROL TO RESIDENT
SEG
CALLS IN QAEXEC.GGG MODULE

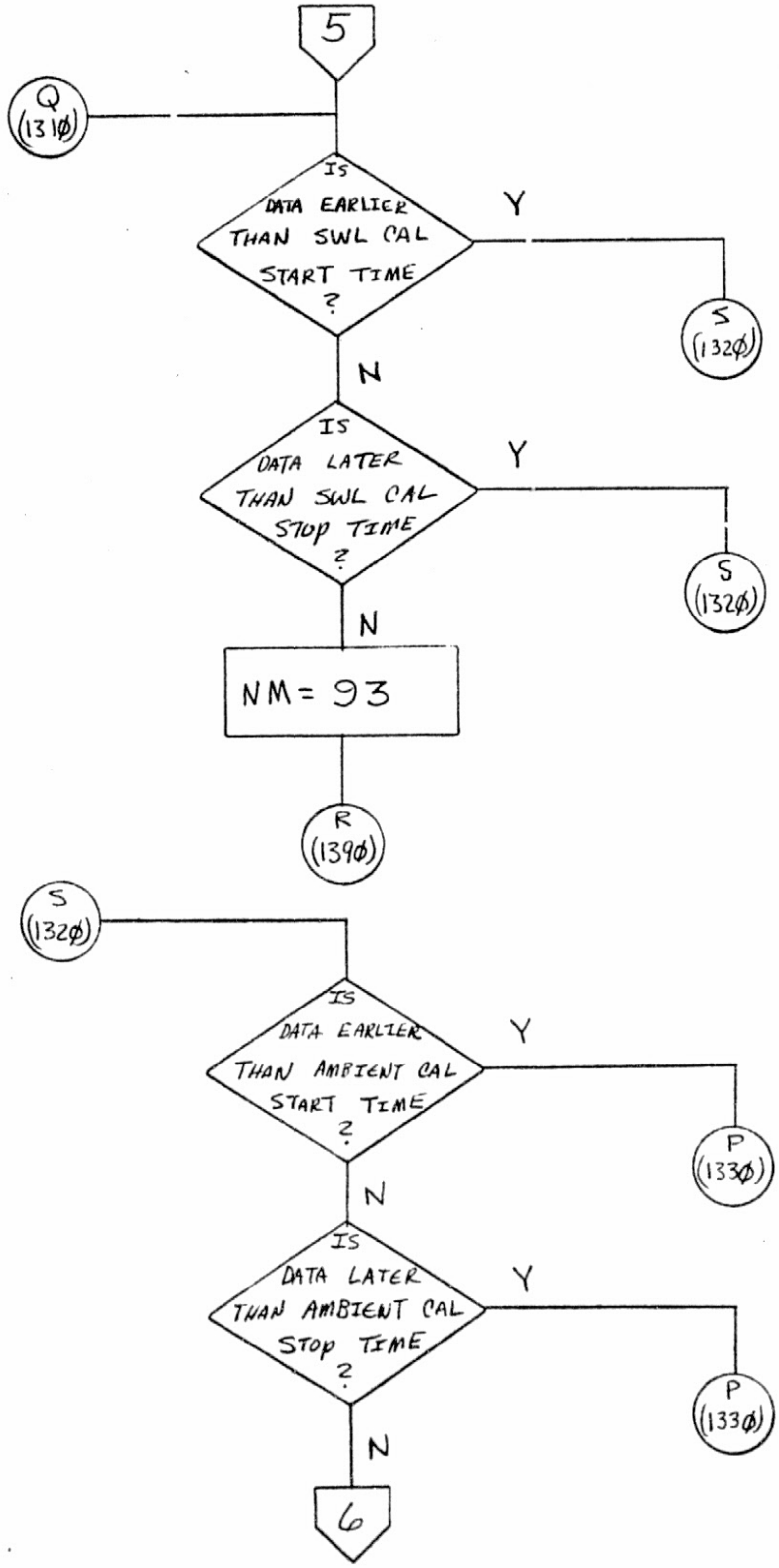

FLDPRO
HAS NO CAL PERIOD BEEN PROCESSED ?
Y
A (1Ø1Ø)
N
SET MAX AND MIN TOLERANCES FOR THERMAL REF. AND DICHROIC TEMP.
IMAX6 = IIN(73)
IMIN6 = IIN(74)
IIN(73) = NDICAL + LMDI
IIN(74) = NDICAL - LMDI
IMAX12 = IIN(85)
IMIN12 = IIN(86)
IIN(85) = NTRCAL + LMTR
IIN(86) = NTRCAL - LMTR
A (1Ø1Ø)
IS DATA OUTSIDE CAL PERIOD TIME SLICE ?
Y
B
N
SET LIMITS TO HANDLE FIRST TEN PARAMETERS:
NMEAS = 82
1

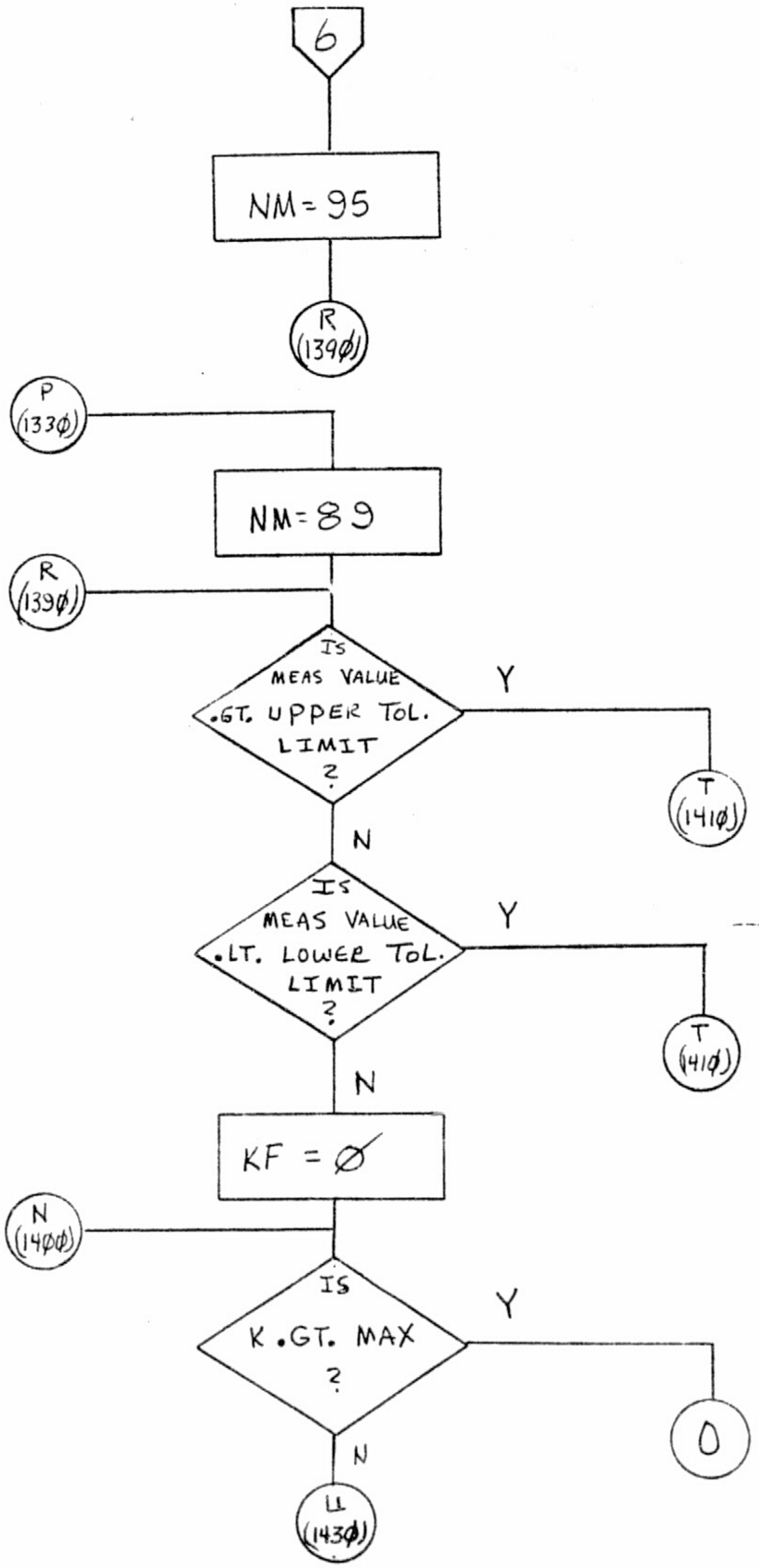

1
B (1020)
SET LIMITS TO HANDLE FIRST EIGHT PARAMETERS:
NMEAS = 78
C (1030)
SET MEAS ARRAY START INDEX (LF):
LF = 231
J=63
J=J+2
IS J > NMEAS ?
G
Y
D (1200)
N
K=1
K=K+1
IS K > MAX?
H
Y
E (1100)
N
IS J.LE.78 ?
Y
F (1040)
N
IS DATA EARLIER THAN CAL PERIOD START TIME ?
Y
E (1100)
N
2

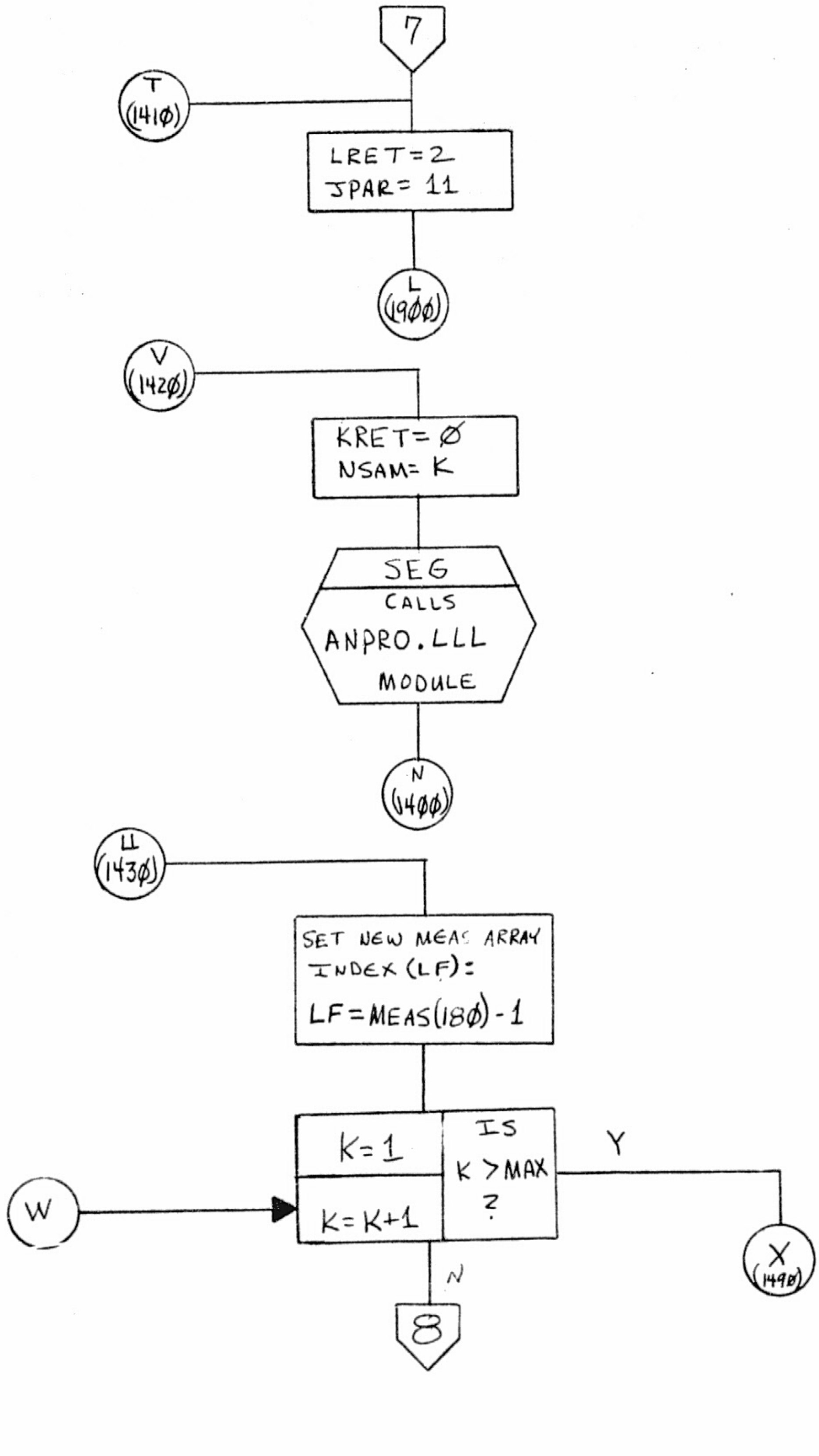

2
IS DATA LATER THAN CAL PERIOD STOP TIME ?
Y
I (115Ø)
F (1Ø4Ø)
N
IS MEAS VALUE .GT. UPPER TOLERANCE ?
Y
J (125Ø)
N
IS MEAS VALUE .LT. LOWER TOLERANCE ?
Y
J (125Ø)
N
KF=Ø
E (11ØØ)
IS K .LE. MAX ?
Y
H
I (115Ø)
N
INCREMENT MEAS ARRAY INDEX: LF=LF+MAX
3

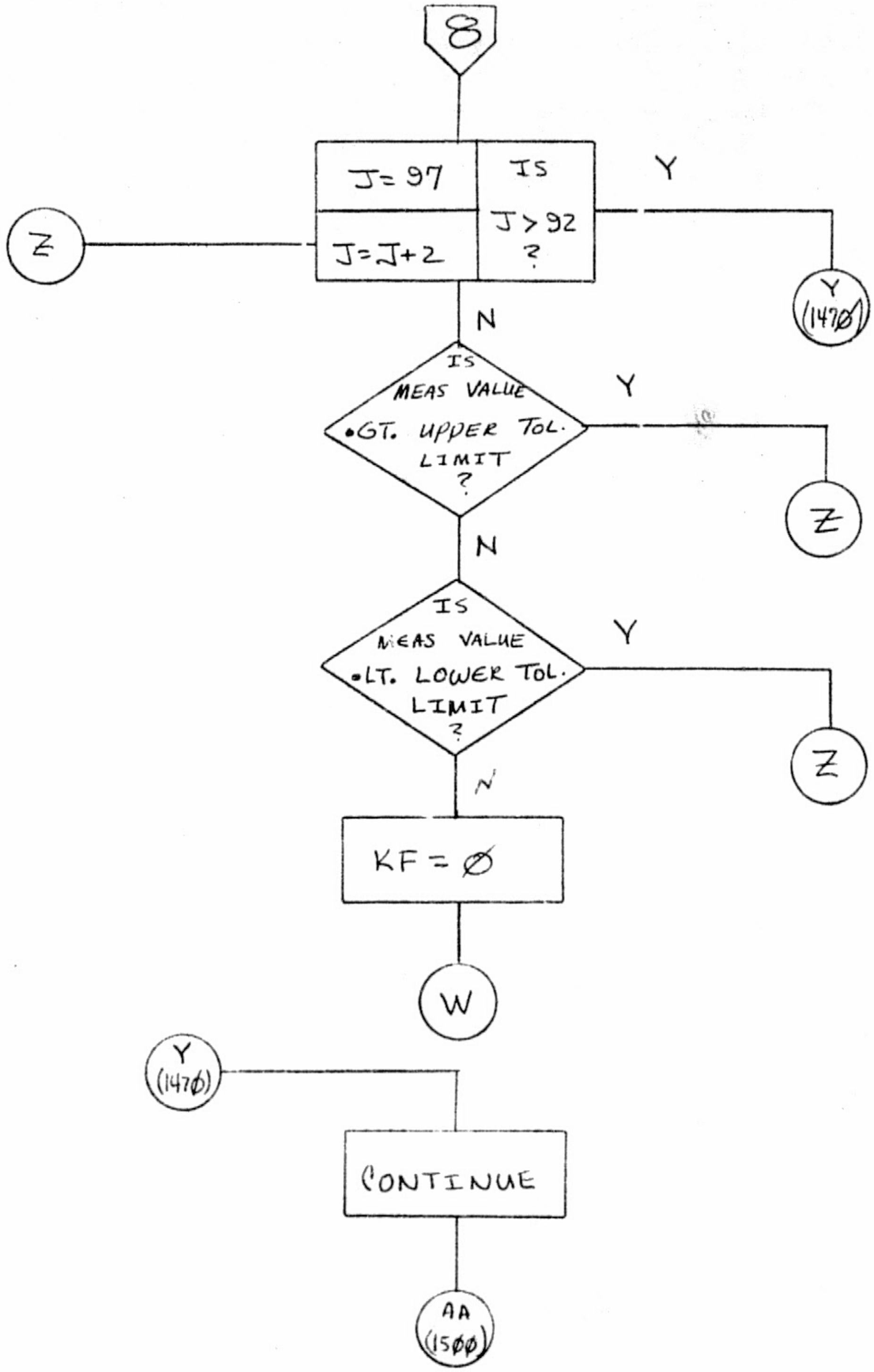

3
D
(12ØØ)
IS
J .LE. NMEAS
?
Y
G
N
K
(13ØØ)
J
(125Ø)
SET RETURN CODE
(LRET) AND ANOMALY
CODE (JPAR):
LRET=1
JPAR=(J-61)/2
L
(19ØØ)
M
(126Ø)
SET SAMPLE NUMBER
(NSAM) AND RETURN
CODE (KRET):
NSAM=K
KRET=Ø
SEG
CALLS
ANPRO. LLL
MODULE
E
(11ØØ)

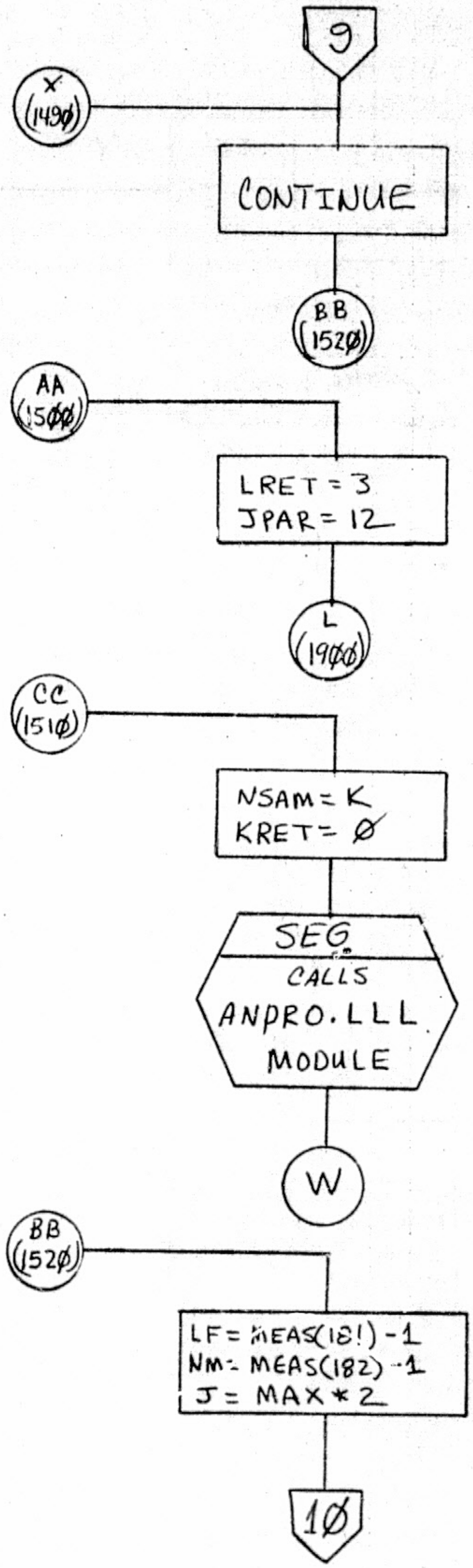

4
K (13ØØ)
RE-SET MEAS ARRAY INDEX (LF):
LF= MEAS(179)-1
K=1
K=K+1
IS K > MAX?
O
Y
N (14ØØ)
N
IS DATA OUTSIDE CAL PERIOD START TIME ?
Y
P (133Ø)
N
IS DATA EARLIER THAN HEATED CAL START TIME ?
Y
Q (131Ø)
N
IS DATA LATER THAN HEATED CAL STOP TIME ?
Y
Q (131Ø)
N
NM= 91
R (139Ø)

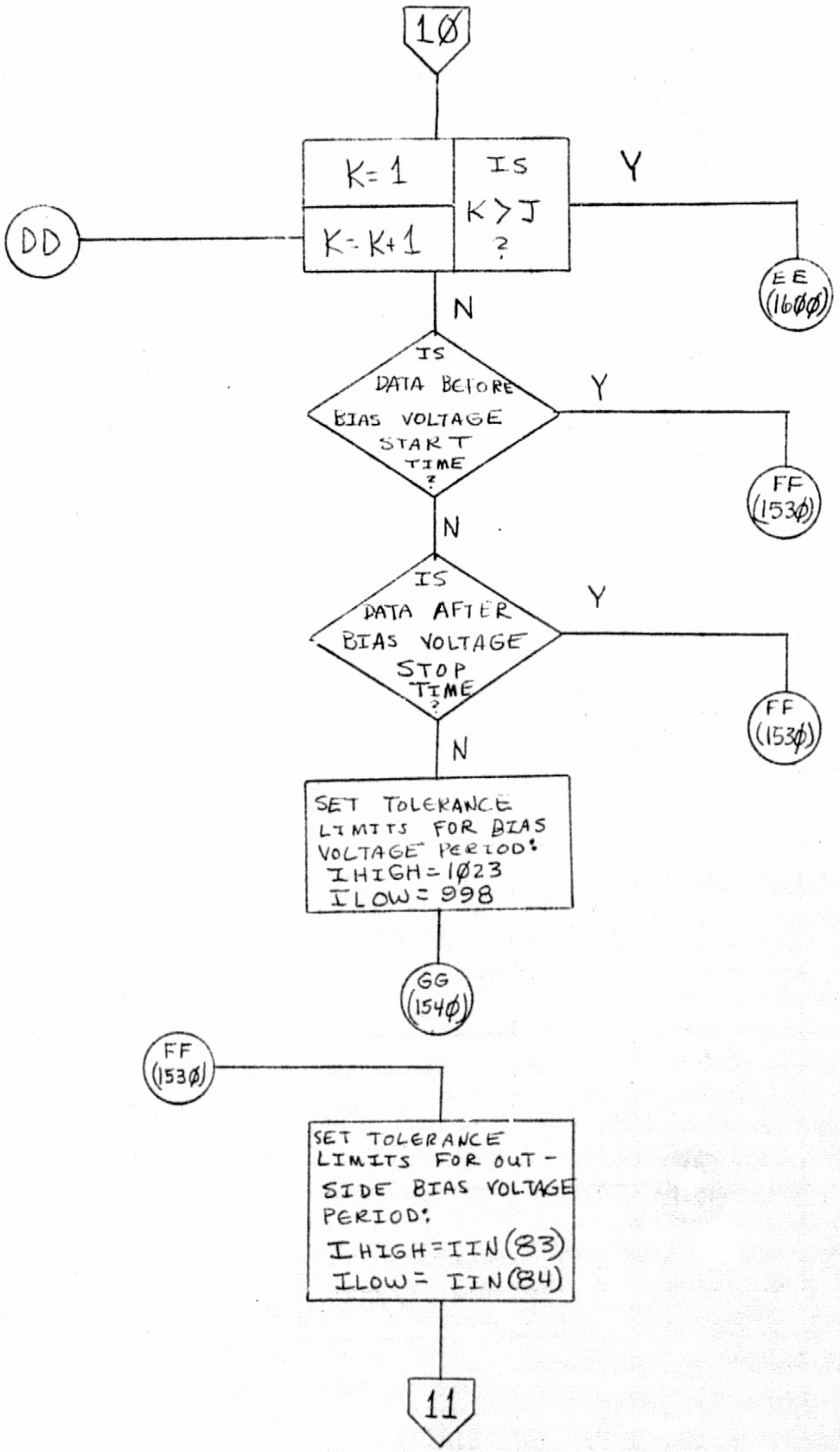

5
Q
(131Ø)
IS DATA EARLIER THAN SWL CAL START TIME ?
Y
S
(132Ø)
N
IS DATA LATER THAN SWL CAL STOP TIME ?
Y
S
(132Ø)
N
NM = 93
R
(139Ø)
S
(132Ø)
IS DATA EARLIER THAN AMBIENT CAL START TIME ?
Y
P
(133Ø)
N
IS DATA LATER THAN AMBIENT CAL STOP TIME ?
Y
P
(133Ø)
N
6

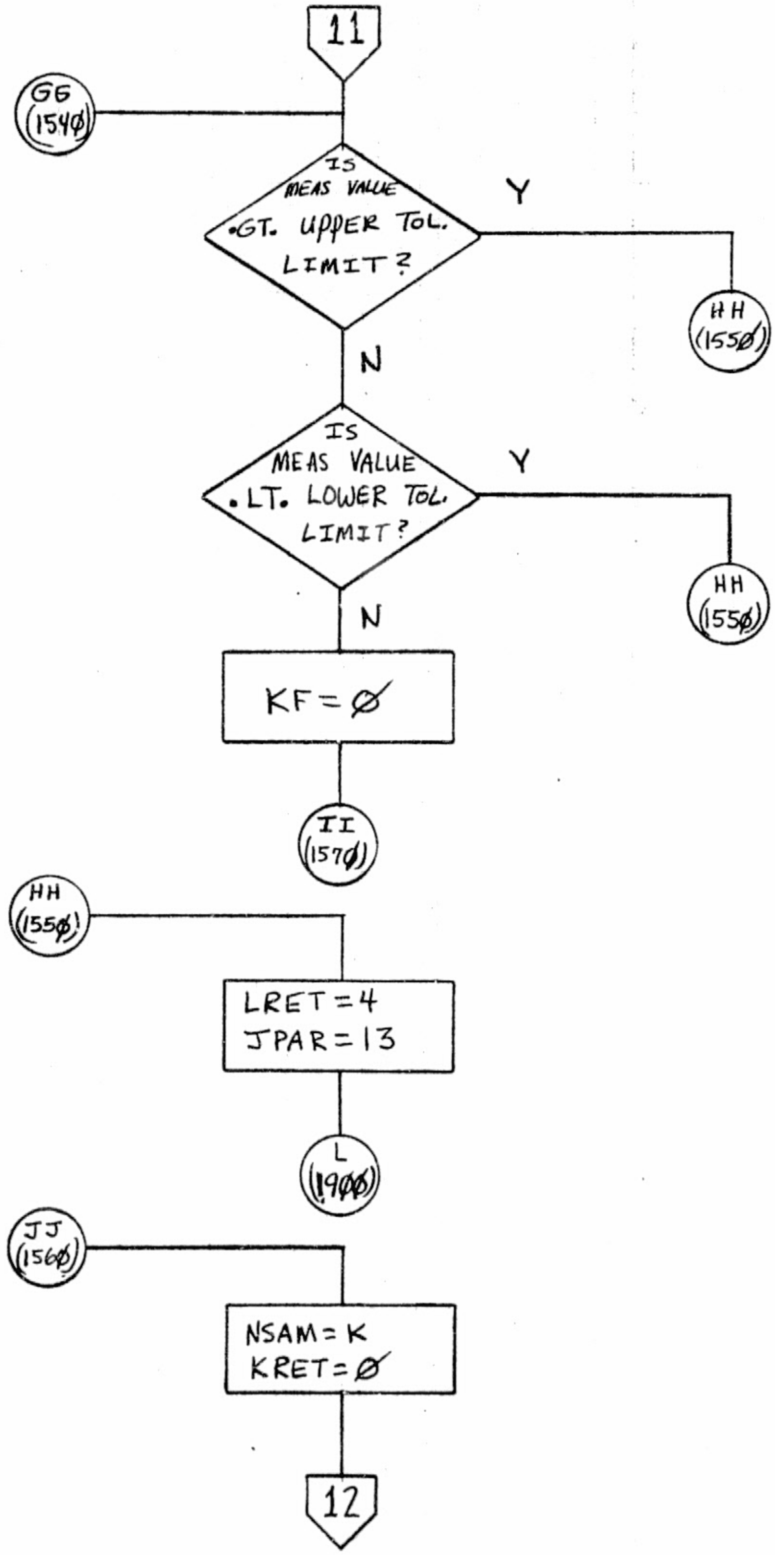

6
NM=95
R (139∅)
P (133∅)
NM=89
R (139∅)
IS MEAS VALUE .GT. UPPER TOL. LIMIT ?
Y
T (141∅)
N
IS MEAS VALUE .LT. LOWER TOL. LIMIT ?
Y
T (141∅)
N
KF = ∅
N (140∅)
IS K .GT. MAX ?
Y
O
N
U (143∅)

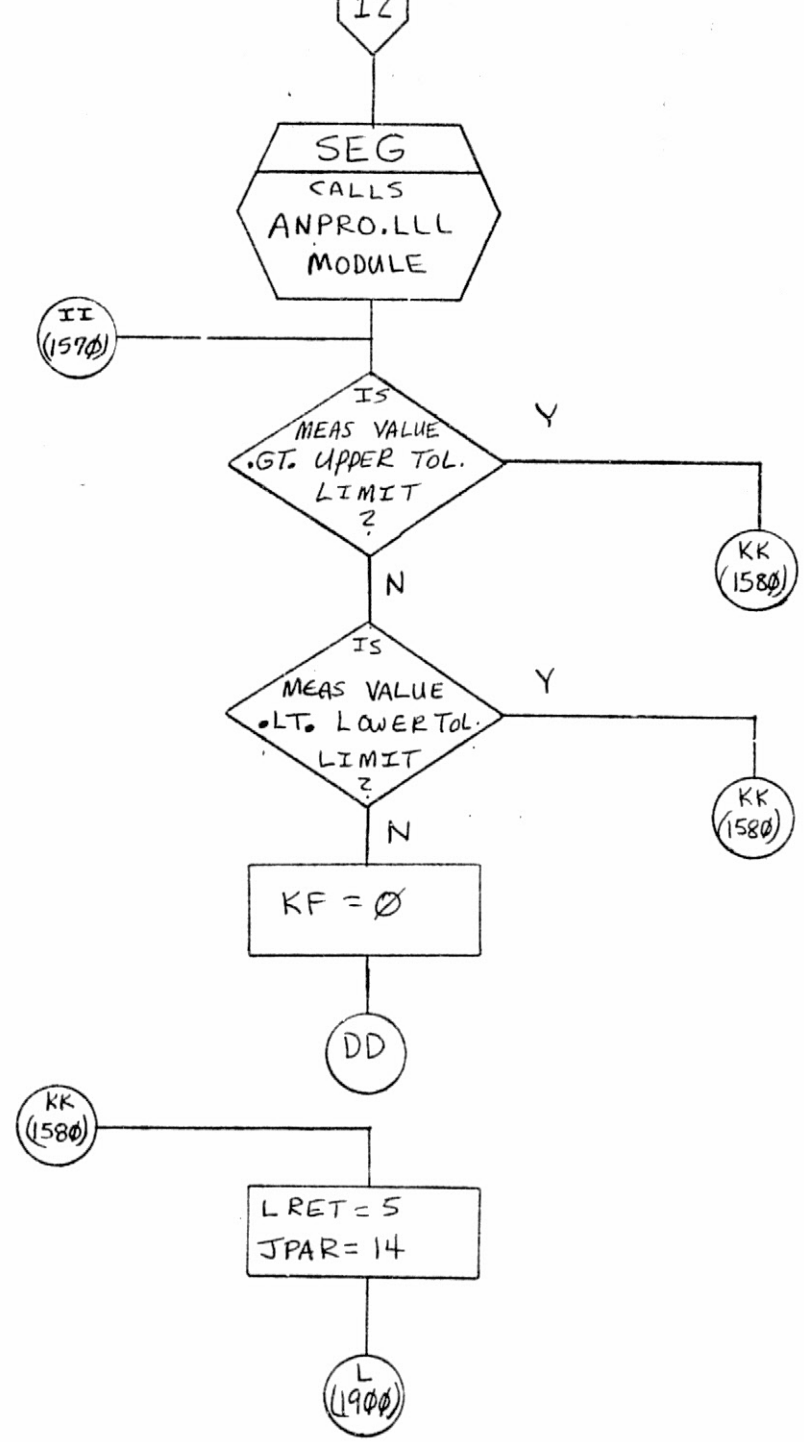

7
T
(1410)
LRET=2
JPAR= 11
L
(1900)
V
(1420)
KRET=0
NSAM= K
SEG
CALLS
ANPRO.LLL
MODULE
N
(1400)
LL
(1430)
SET NEW MEAS ARRAY
INDEX (LF):
LF=MEAS(180)-1
K=1
K=K+1
IS
K >MAX
?
Y
N
W
X
(1490)
8

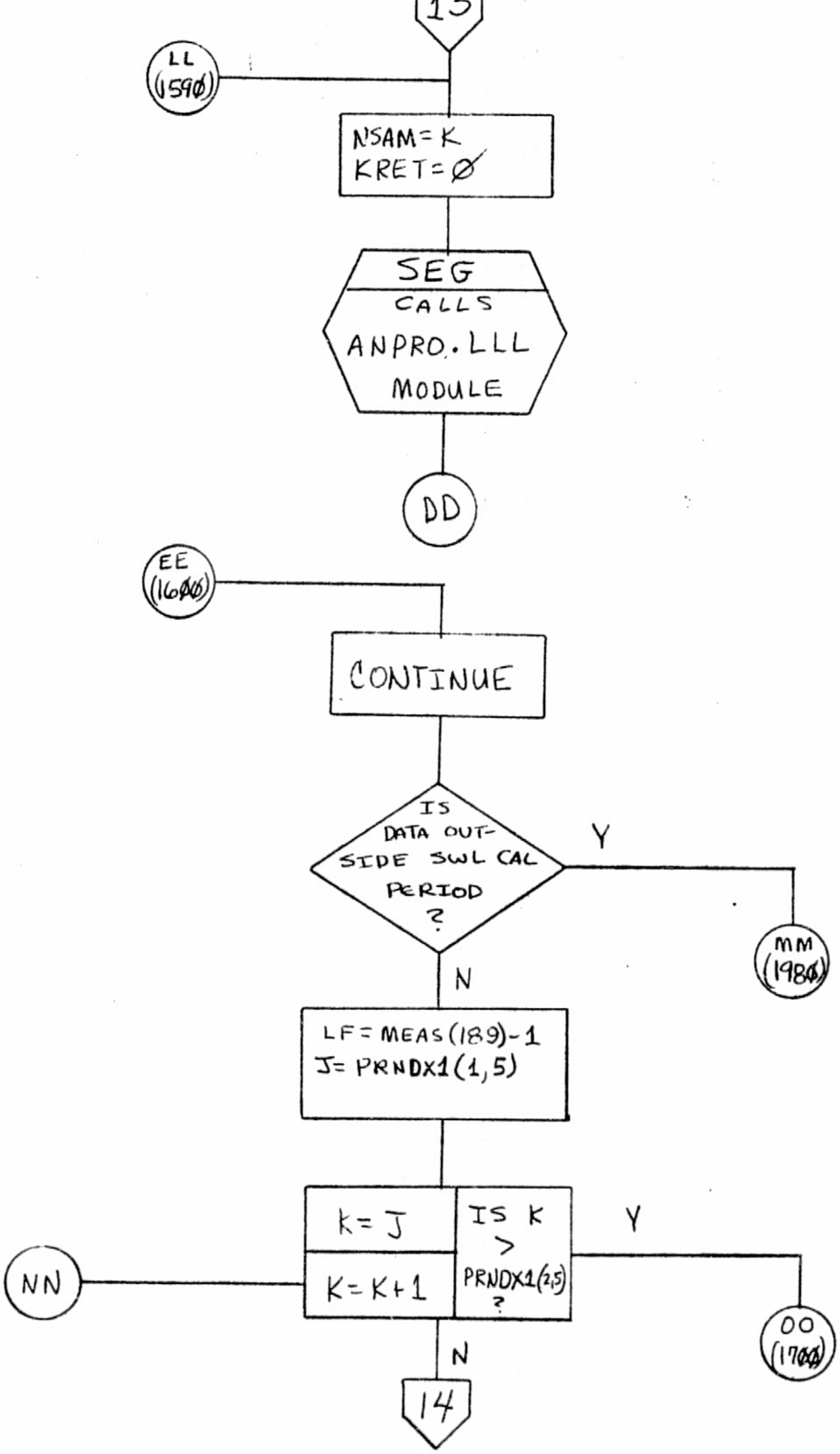

8
J = 97
IS J > 92 ?
Y
J = J + 2
Z
Y (147Ø)
N
IS MEAS VALUE .GT. UPPER TOL. LIMIT ?
Y
Z
N
IS MEAS VALUE .LT. LOWER TOL. LIMIT ?
Y
Z
N
KF = Ø
W
Y (147Ø)
CONTINUE
AA (15ØØ)

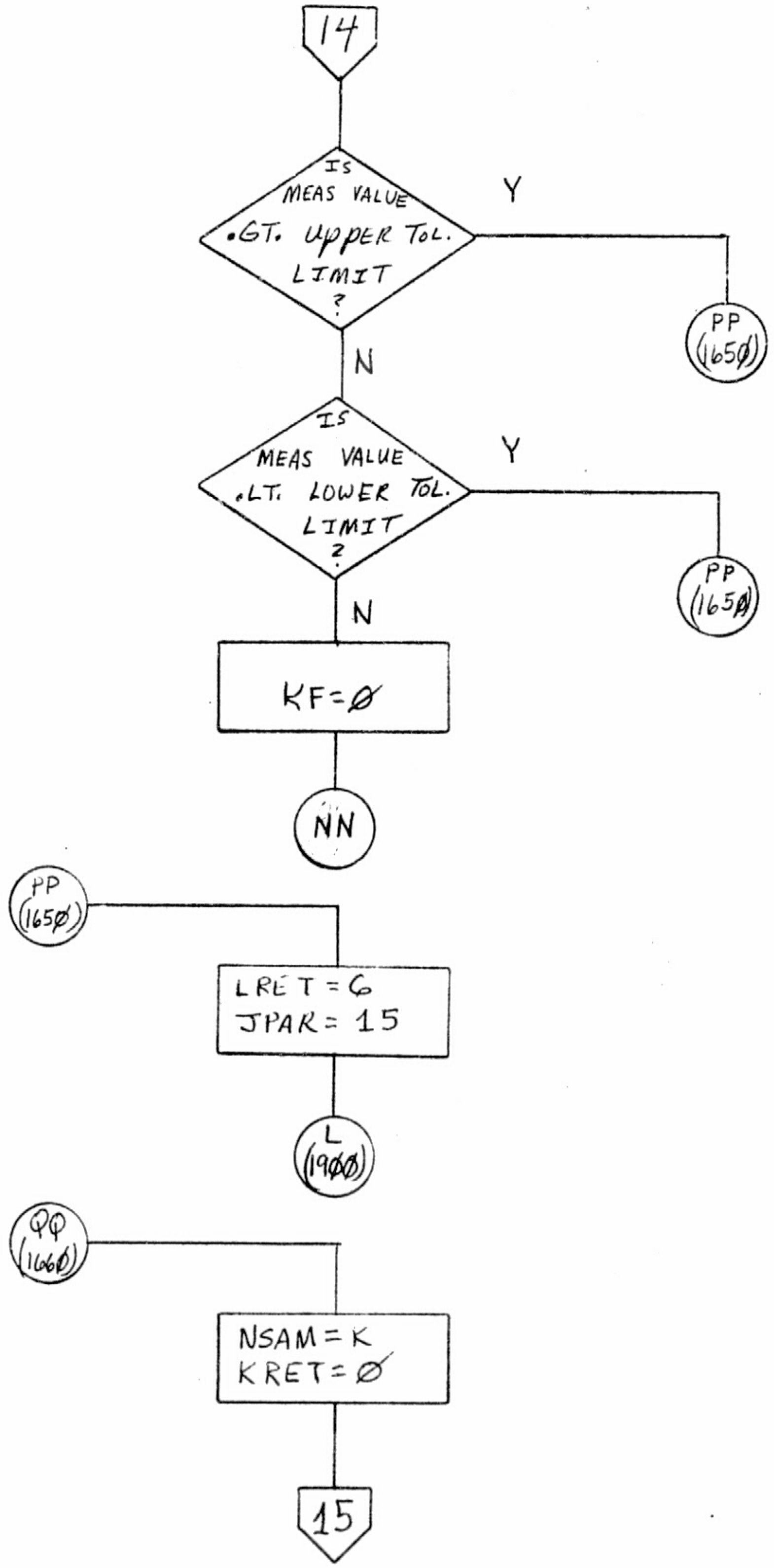

9
X
(149Ø)
CONTINUE
BB
(152Ø)
AA
(150Ø)
LRET = 3
JPAR = 12
L
(190Ø)
CC
(151Ø)
NSAM = K
KRET = Ø
SEG
CALLS
ANPRO.LLL
MODULE
W
BB
(152Ø)
LF = MEAS(181) - 1
NM = MEAS(182) - 1
J = MAX * 2
1Ø

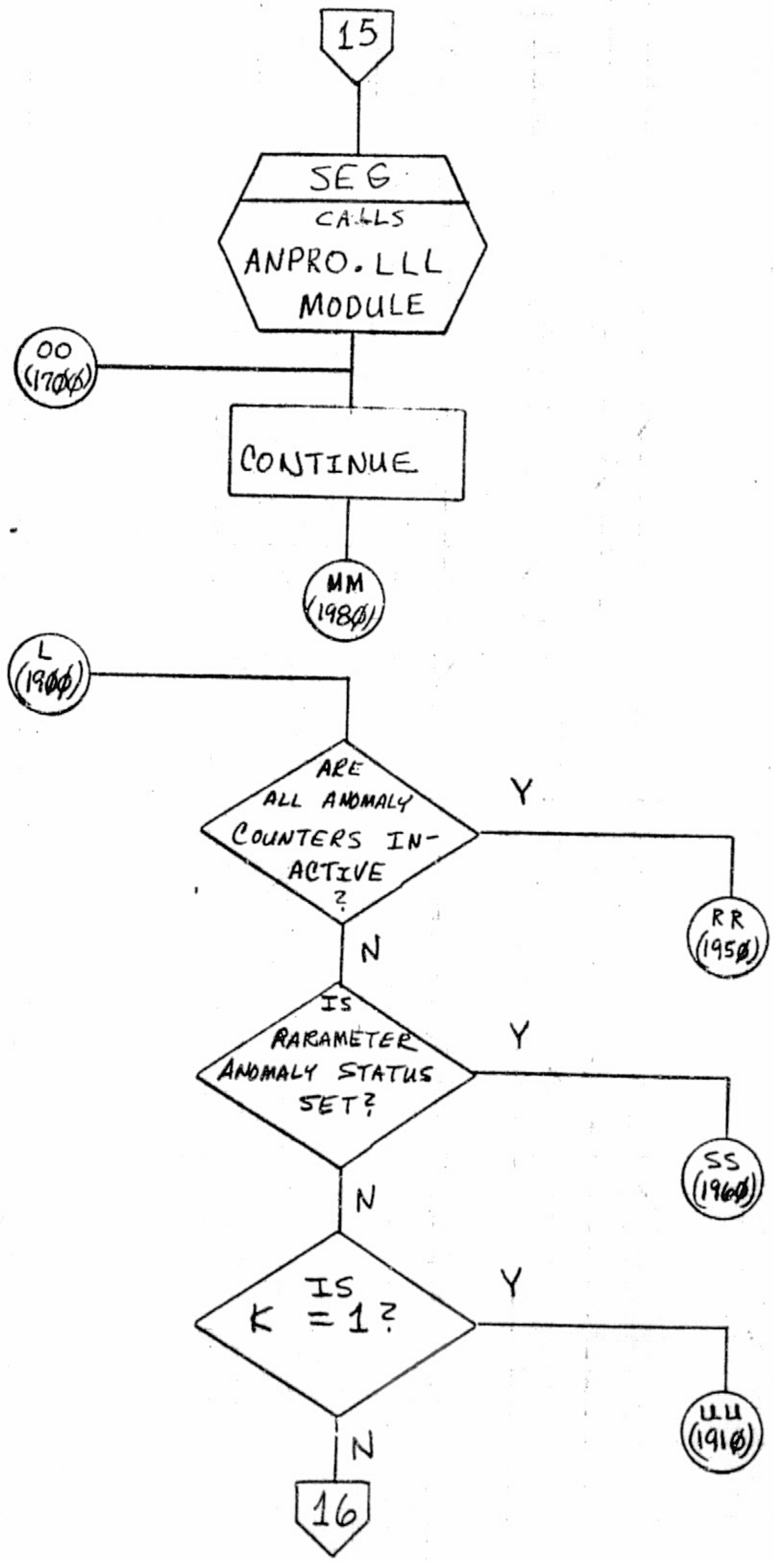

10
K = 1
K = K+1
IS K > J ?
Y
N
DD
EE (16ØØ)
IS DATA BEFORE BIAS VOLTAGE START TIME ?
Y
N
FF (153Ø)
IS DATA AFTER BIAS VOLTAGE STOP TIME ?
Y
N
FF (153Ø)
SET TOLERANCE LIMITS FOR BIAS VOLTAGE PERIOD:
IHIGH = 1Ø23
ILOW = 998
GG (154Ø)
FF (153Ø)
SET TOLERANCE LIMITS FOR OUTSIDE BIAS VOLTAGE PERIOD:
IHIGH = IIN(83)
ILOW = IIN(84)
11

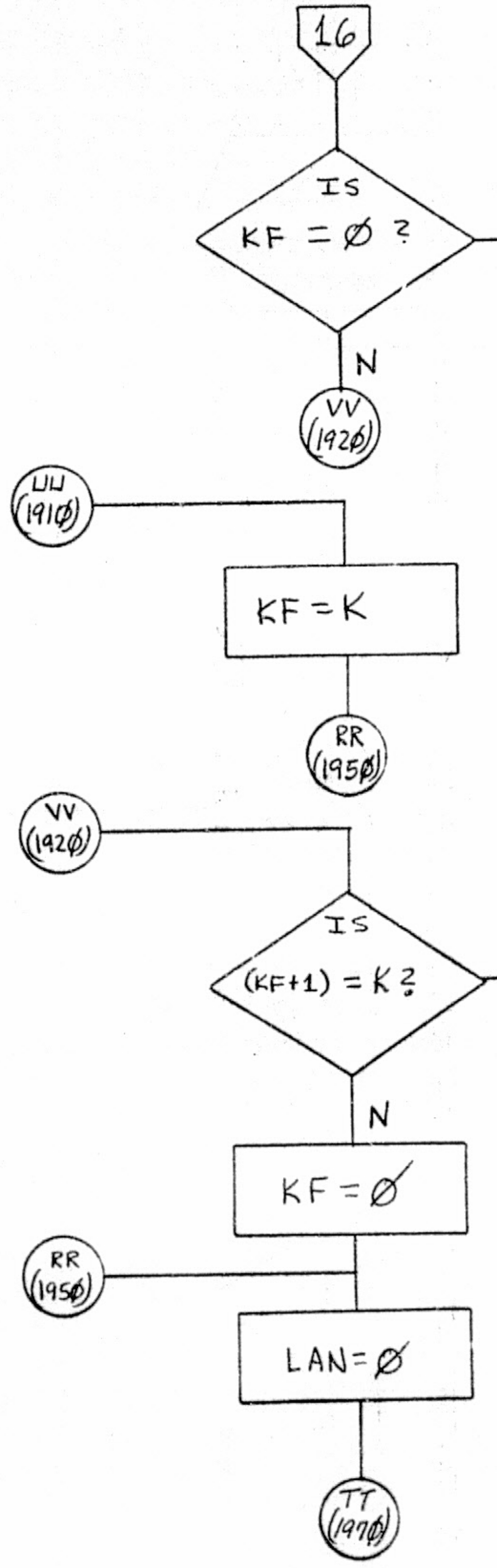

11
GG (154Ø)
IS MEAS VALUE .GT. UPPER TOL. LIMIT?
Y
HH (155Ø)
N
IS MEAS VALUE .LT. LOWER TOL. LIMIT?
Y
HH (155Ø)
N
KF=Ø
II (157Ø)
HH (155Ø)
LRET=4
JPAR=13
L (19ØØ)
JJ (156Ø)
NSAM=K
KRET=Ø
12

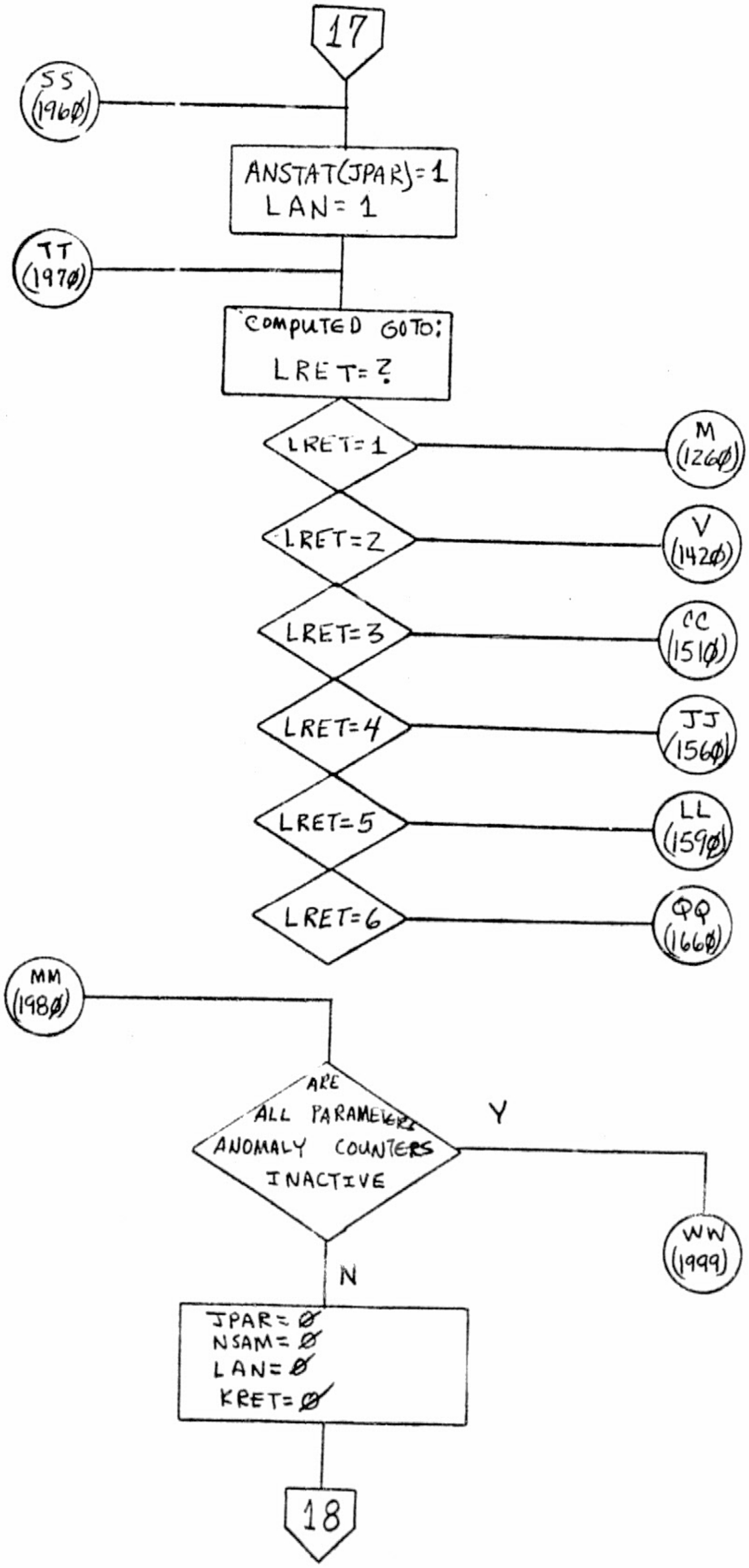

12
SEG
CALLS
ANPRO.LLL
MODULE
II
(157∅)
IS
MEAS VALUE
.GT. UPPER TOL.
LIMIT
?
Y
KK
(158∅)
N
IS
MEAS VALUE
.LT. LOWER TOL.
LIMIT
?
Y
KK
(158∅)
N
KF = ∅
DD
KK
(158∅)
LRET = 5
JPAR = 14
L
(19∅∅)

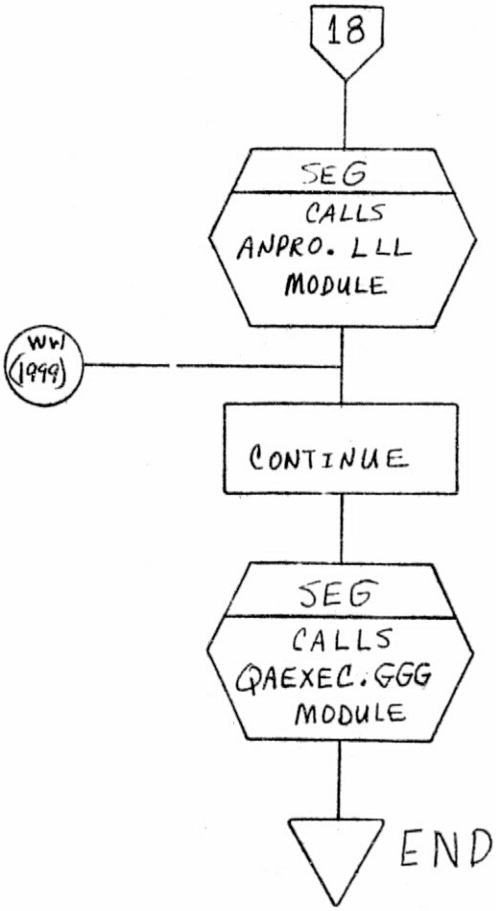

13
LL
(1590)
NSAM=K
KRET=Ø
SEG
CALLS
ANPRO.LLL
MODULE
DD
EE
(1600)
CONTINUE
IS
DATA OUT-
SIDE SWL CAL
PERIOD
?
Y
MM
(1980)
N
LF = MEAS (189)-1
J= PRNDX1(1,5)
K= J
K=K+1
IS K
>
PRNDX1(2,5)
?
Y
NN
OO
(1700)
N
14

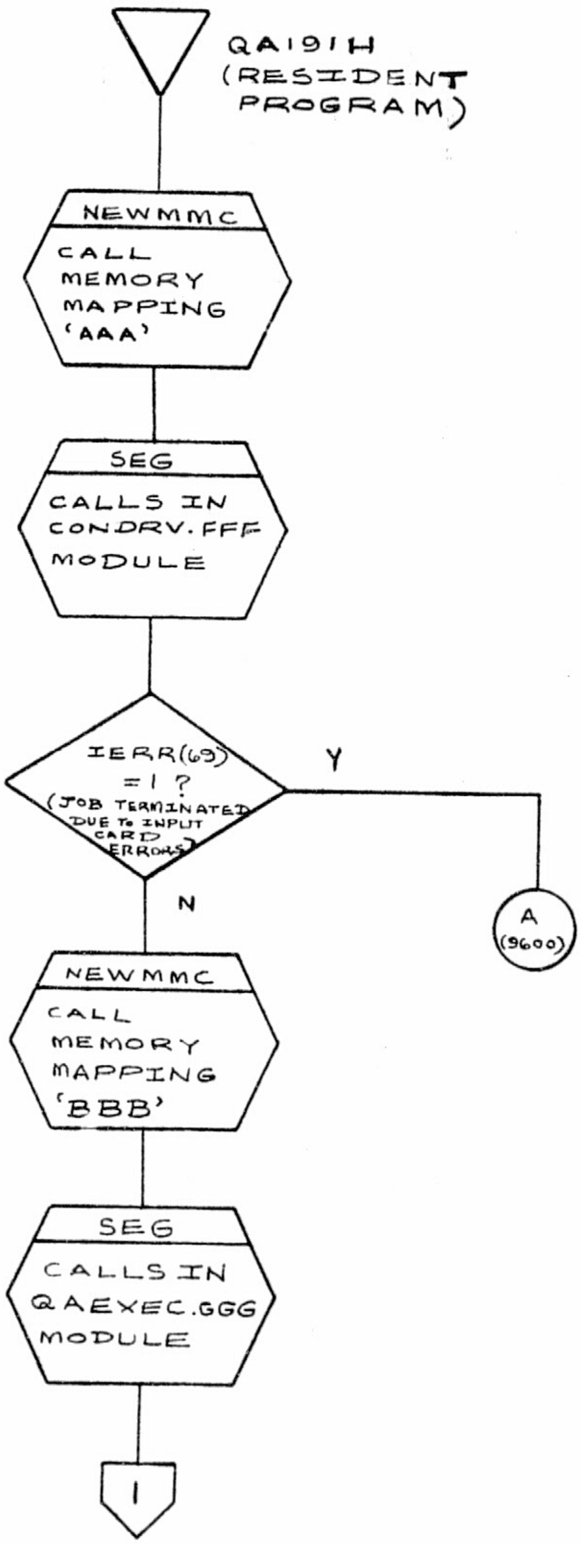

14
IS MEAS VALUE .GT. UPPER TOL. LIMIT ?
Y
PP (165Ø)
N
IS MEAS VALUE .LT. LOWER TOL. LIMIT ?
Y
PP (165Ø)
N
KF=Ø
NN
PP (165Ø)
LRET = 6
JPAR = 15
L (19ØØ)
QQ (166Ø)
NSAM = K
KRET = Ø
15

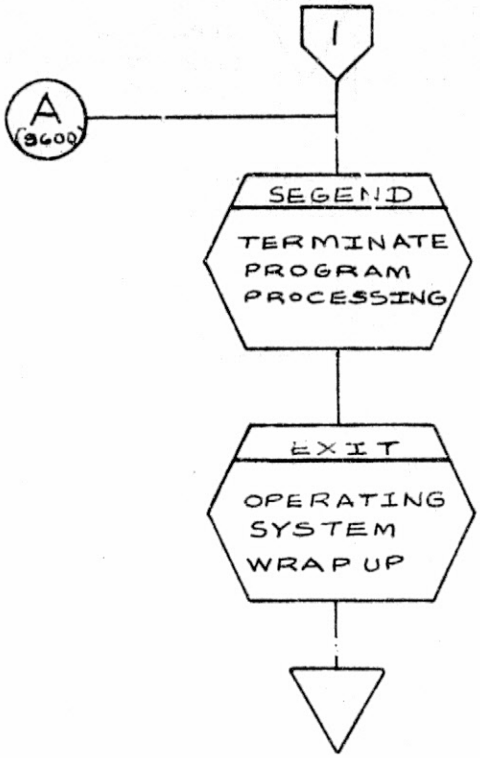

15
SEG
CALLS
ANPRO.LLL
MODULE
OO
(17ØØ)
CONTINUE
MM
(198Ø)
L
(19ØØ)
ARE
ALL ANOMALY
COUNTERS IN-
ACTIVE
?
Y
RR
(195Ø)
N
IS
PARAMETER
ANOMALY STATUS
SET?
Y
SS
(196Ø)
N
IS
K = 1?
Y
UU
(191Ø)
N
16

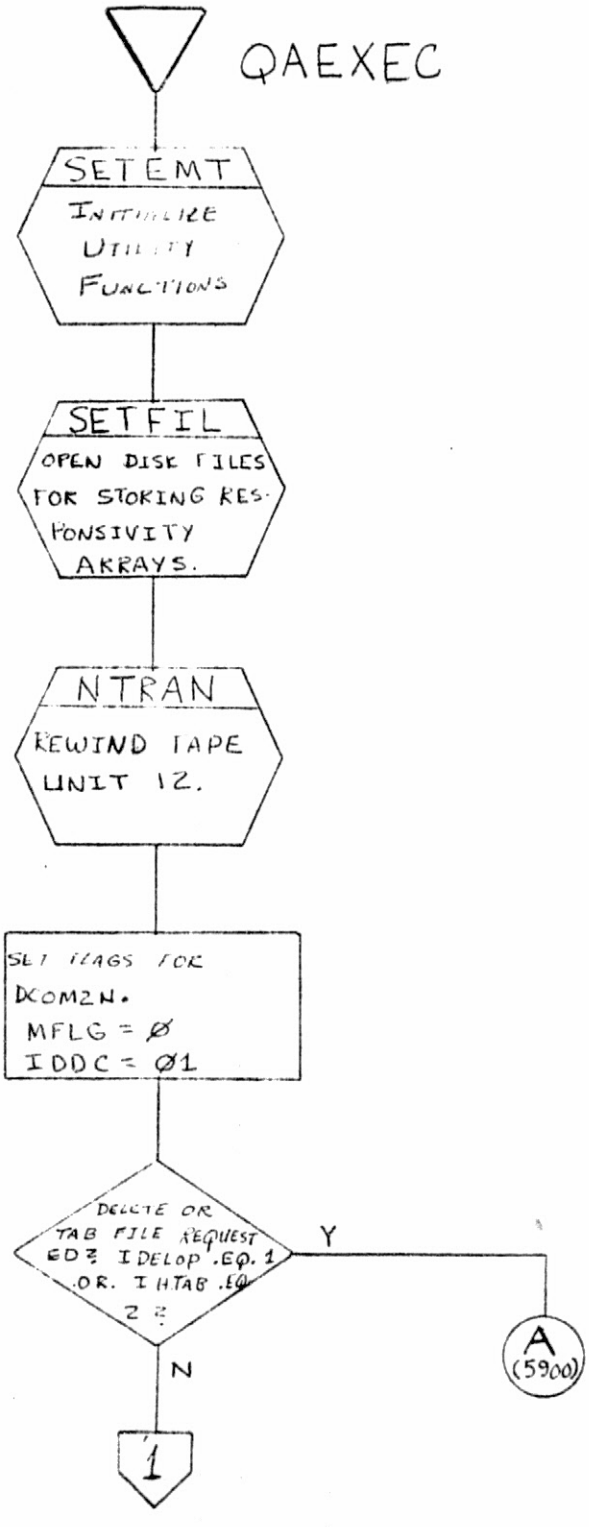

16
IS
KF = Ø ?
Y
N
UU
(191Ø)
VV
(192Ø)
UU
(191Ø)
KF = K
RR
(195Ø)
VV
(192Ø)
IS
(KF+1) = K ?
Y
SS
(196Ø)
N
KF = Ø
RR
(195Ø)
LAN = Ø
TT
(197Ø)

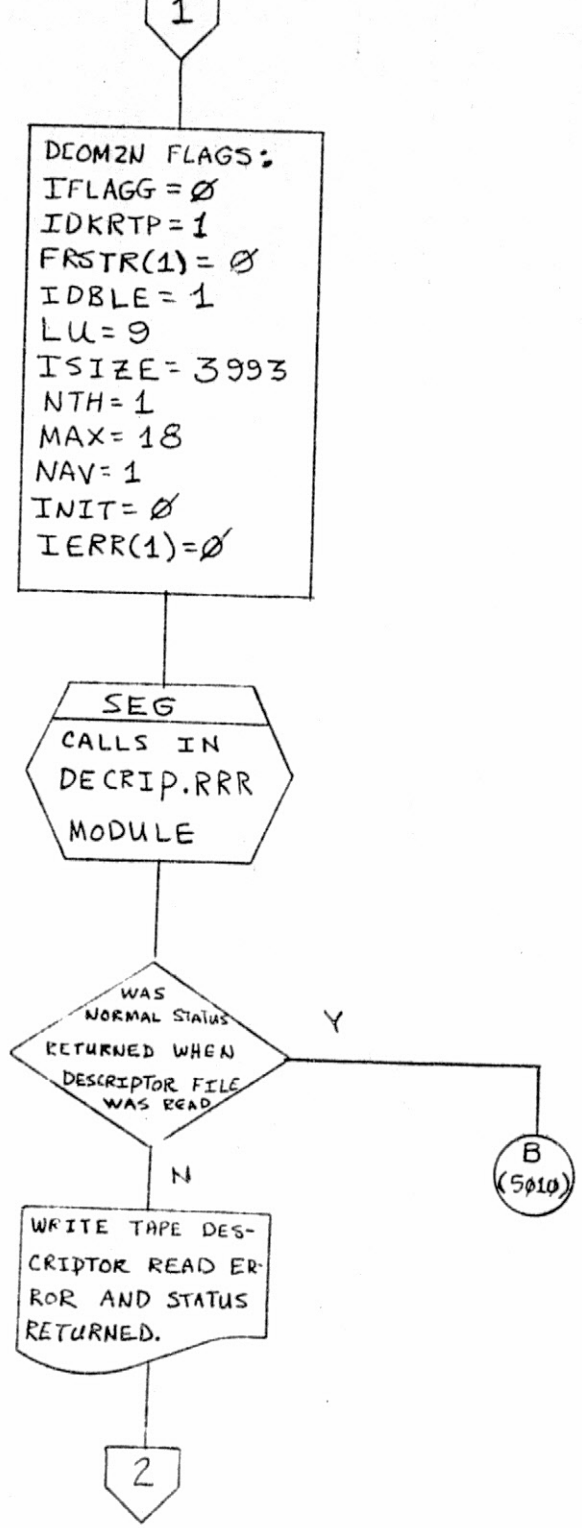

17
SS (196Ø)
ANSTAT(JPAR)=1
LAN=1
TT (197Ø)
COMPUTED GOTO:
LRET=?
LRET=1
M (126Ø)
LRET=2
V (142Ø)
LRET=3
CC (151Ø)
LRET=4
JJ (156Ø)
LRET=5
LL (159Ø)
LRET=6
QQ (166Ø)
MM (198Ø)
ARE ALL PARAMETERS ANOMALY COUNTERS INACTIVE
Y
WW (1999)
N
JPAR=Ø
NSAM=Ø
LAN=Ø
KRET=Ø
18

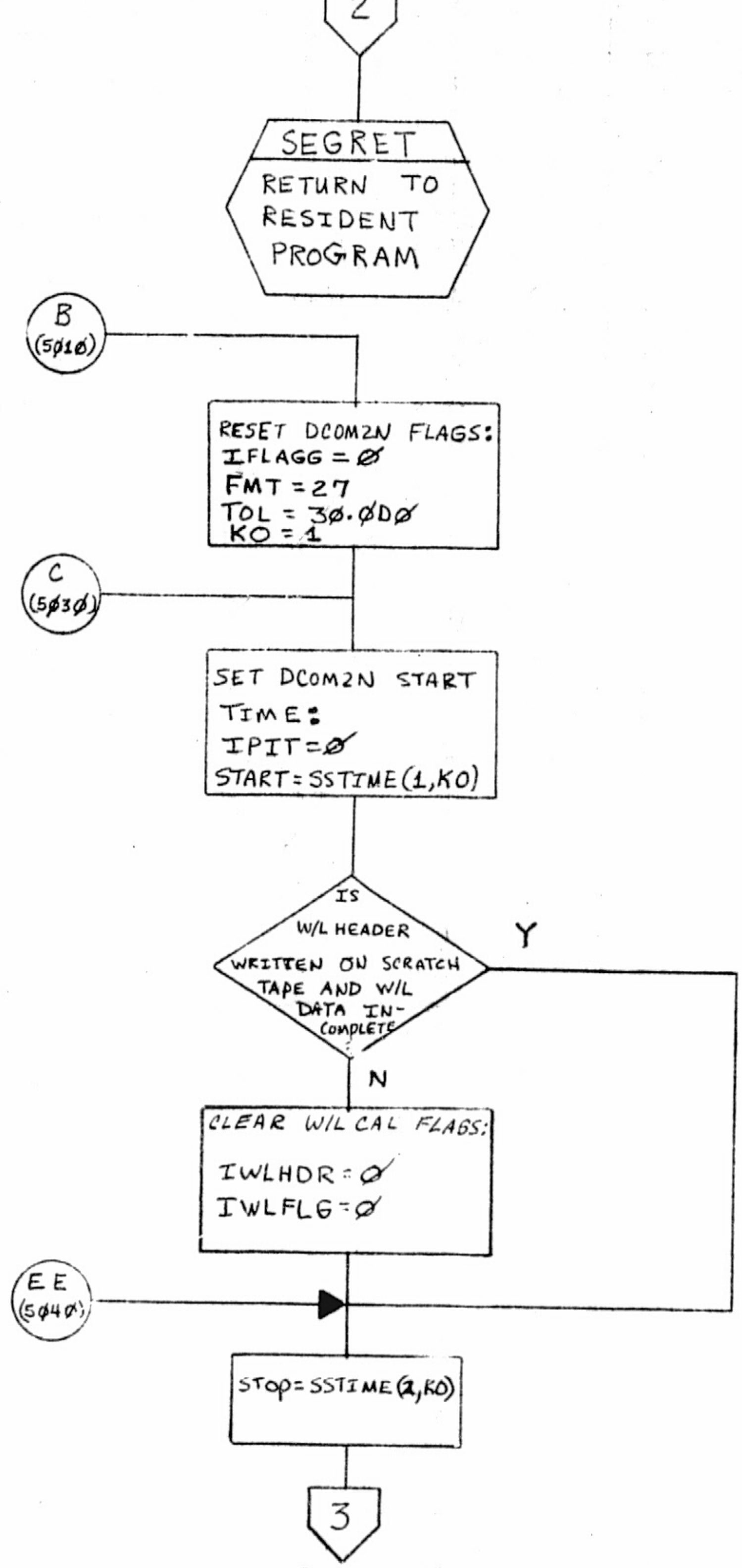

18
SEG
CALLS
ANPRO. LLL
MODULE
WW
(1999)
CONTINUE
SEG
CALLS
QAEXEC.GGG
MODULE
END

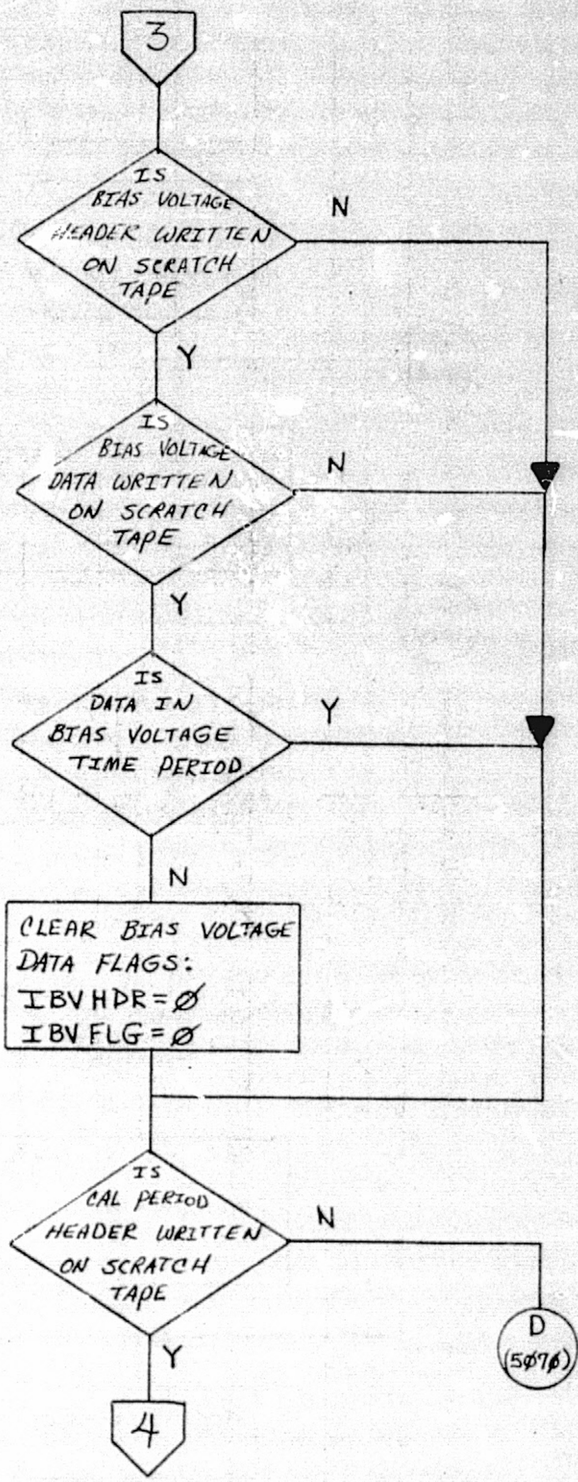

QA191H
(RESIDENT
PROGRAM)
NEWMMC
CALL
MEMORY
MAPPING
'AAA'
SEG
CALLS IN
CONDRV.FFF
MODULE
IERR(69)
=1 ?
(JOB TERMINATED
DUE TO INPUT
CARD
ERRORS)
Y
N
A
(9600)
NEWMMC
CALL
MEMORY
MAPPING
'BBB'
SEG
CALLS IN
QAEXEC.GGG
MODULE
1

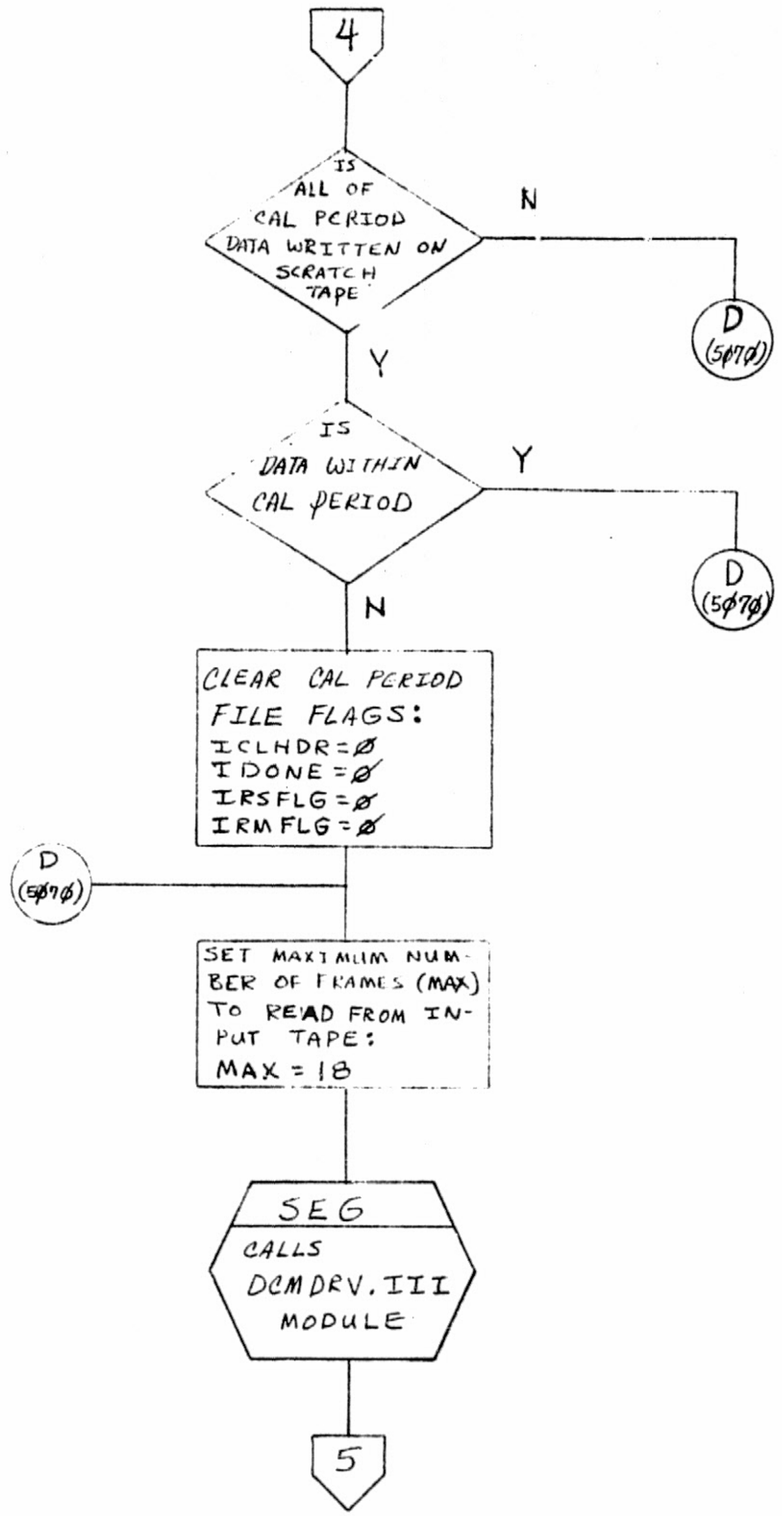

1
A
9600
SEGEND
TERMINATE PROGRAM PROCESSING
EXIT
OPERATING SYSTEM WRAP UP

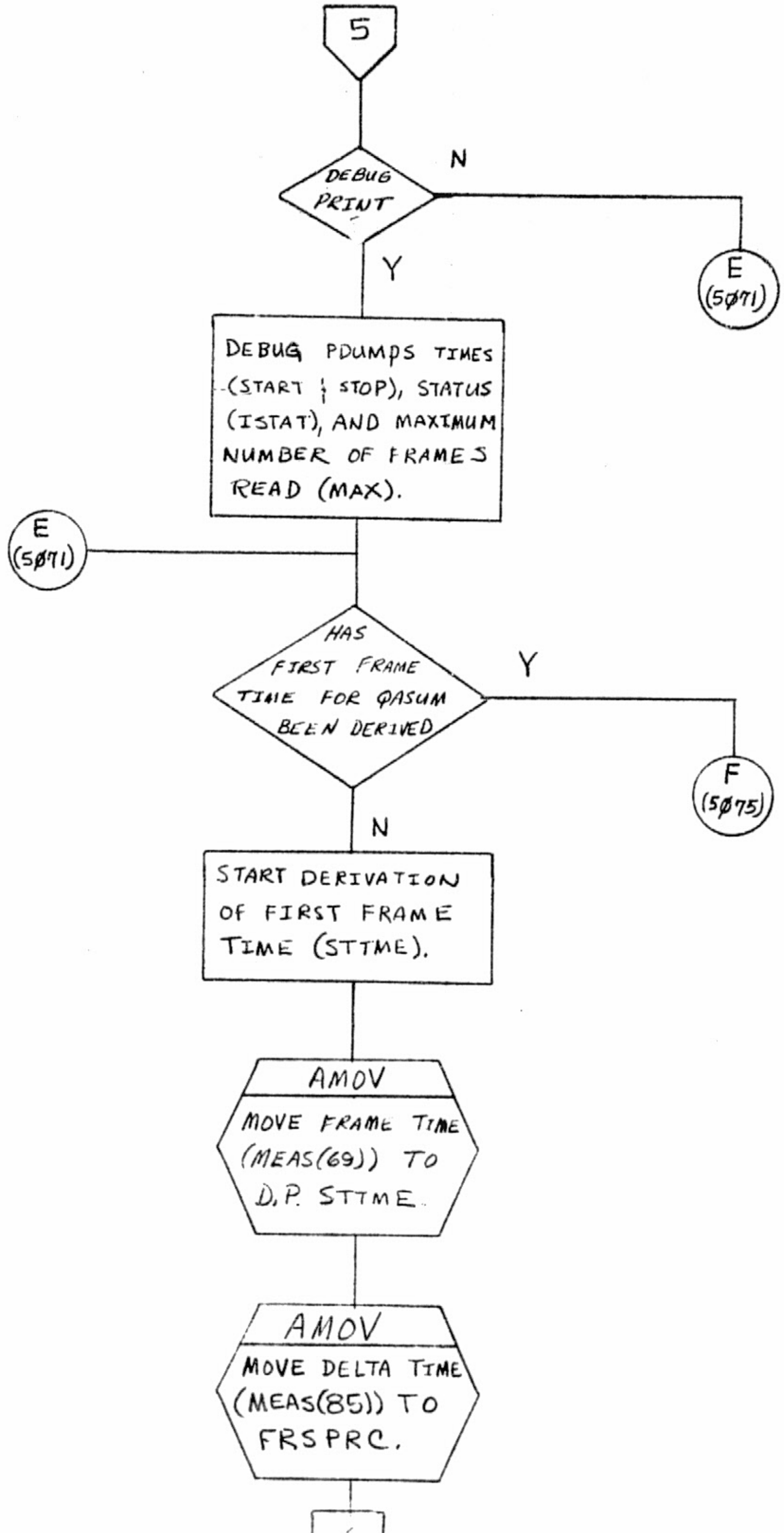

QAEXEC
SETEMT
INITIALIZE UTILITY FUNCTIONS
SETFIL
OPEN DISK FILES FOR STORING RES-PONSIVITY ARRAYS.
NTRAN
REWIND TAPE UNIT 12.
SET FLAGS FOR DCOM2N.
MFLG = Ø
IDDC = Ø1
DELETE OR TAB FILE REQUEST ED? IDELOP .EQ. 1 .OR. IHTAB .EQ. 2 ?
Y
N
A (5900)
1

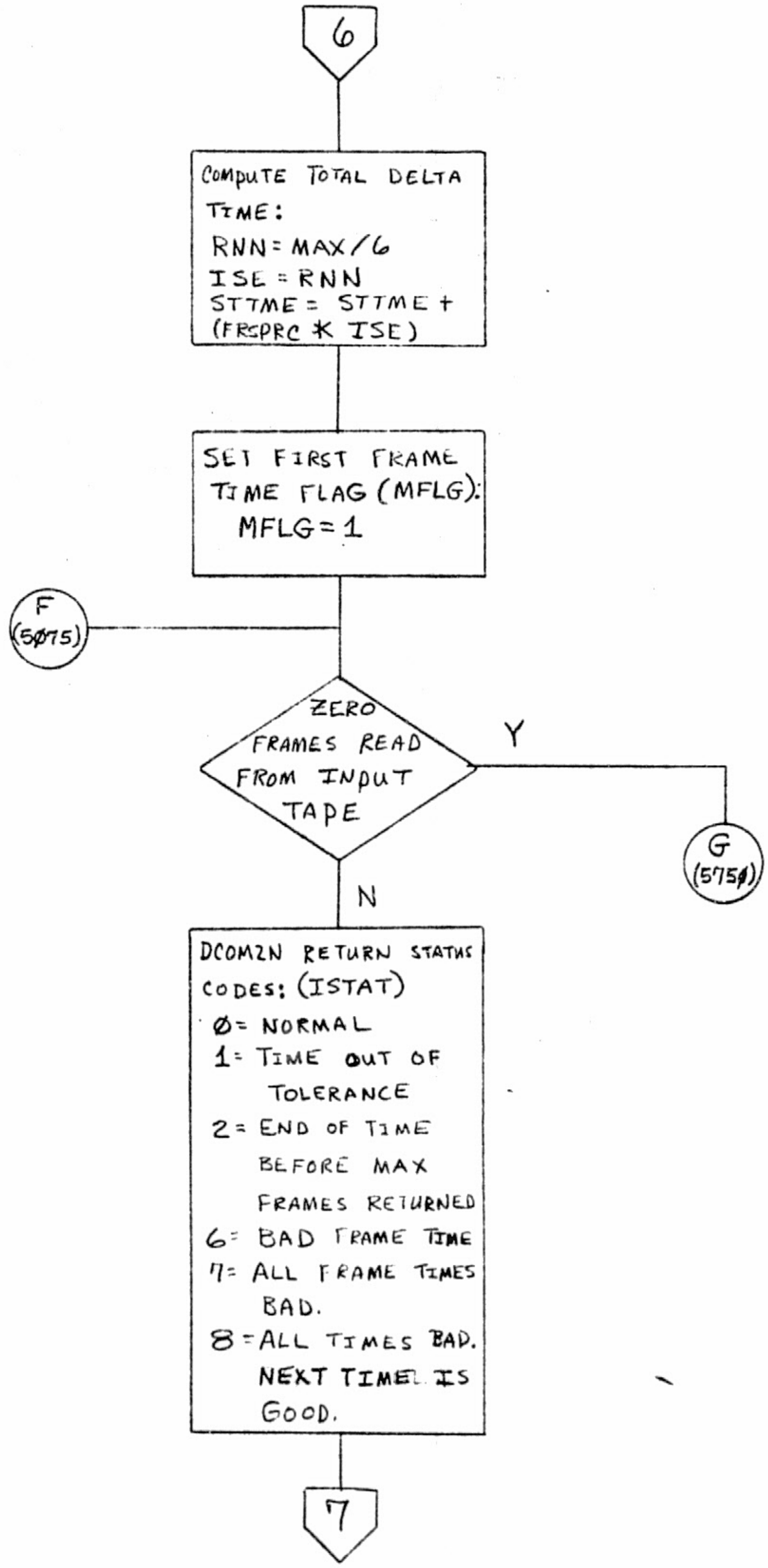

1
DCOM2N FLAGS:
IFLAGG = Ø
IDKRTP = 1
FRSTR(1) = Ø
IDBLE = 1
LU = 9
ISIZE = 3993
NTH = 1
MAX = 18
NAV = 1
INIT = Ø
IERR(1) = Ø
SEG
CALLS IN DECRIP.RRR MODULE
WAS NORMAL STATUS RETURNED WHEN DESCRIPTOR FILE WAS READ
Y
B (5Ø1Ø)
N
WRITE TAPE DESCRIPTOR READ ERROR AND STATUS RETURNED.
2

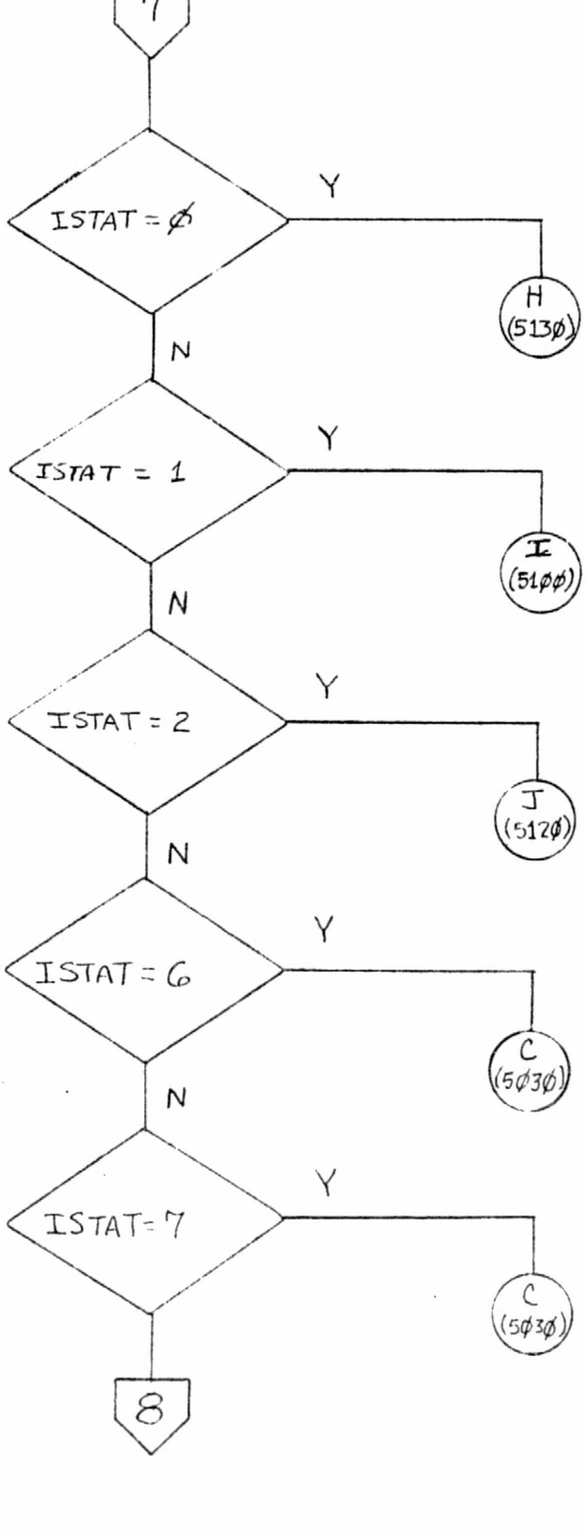

2
SEGRET
RETURN TO RESIDENT PROGRAM
B (5Ø1Ø)
RESET DCOM2N FLAGS:
IFLAGG = Ø
FMT = 27
TOL = 3Ø.ØDØ
KO = 1
C (5Ø3Ø)
SET DCOM2N START TIME:
IPIT=Ø
START=SSTIME(1,KO)
IS W/L HEADER WRITTEN ON SCRATCH TAPE AND W/L DATA INCOMPLETE
Y
N
CLEAR W/L CAL FLAGS:
IWLHDR=Ø
IWLFLG=Ø
EE (5Ø4Ø)
STOP=SSTIME(2,KO)
3

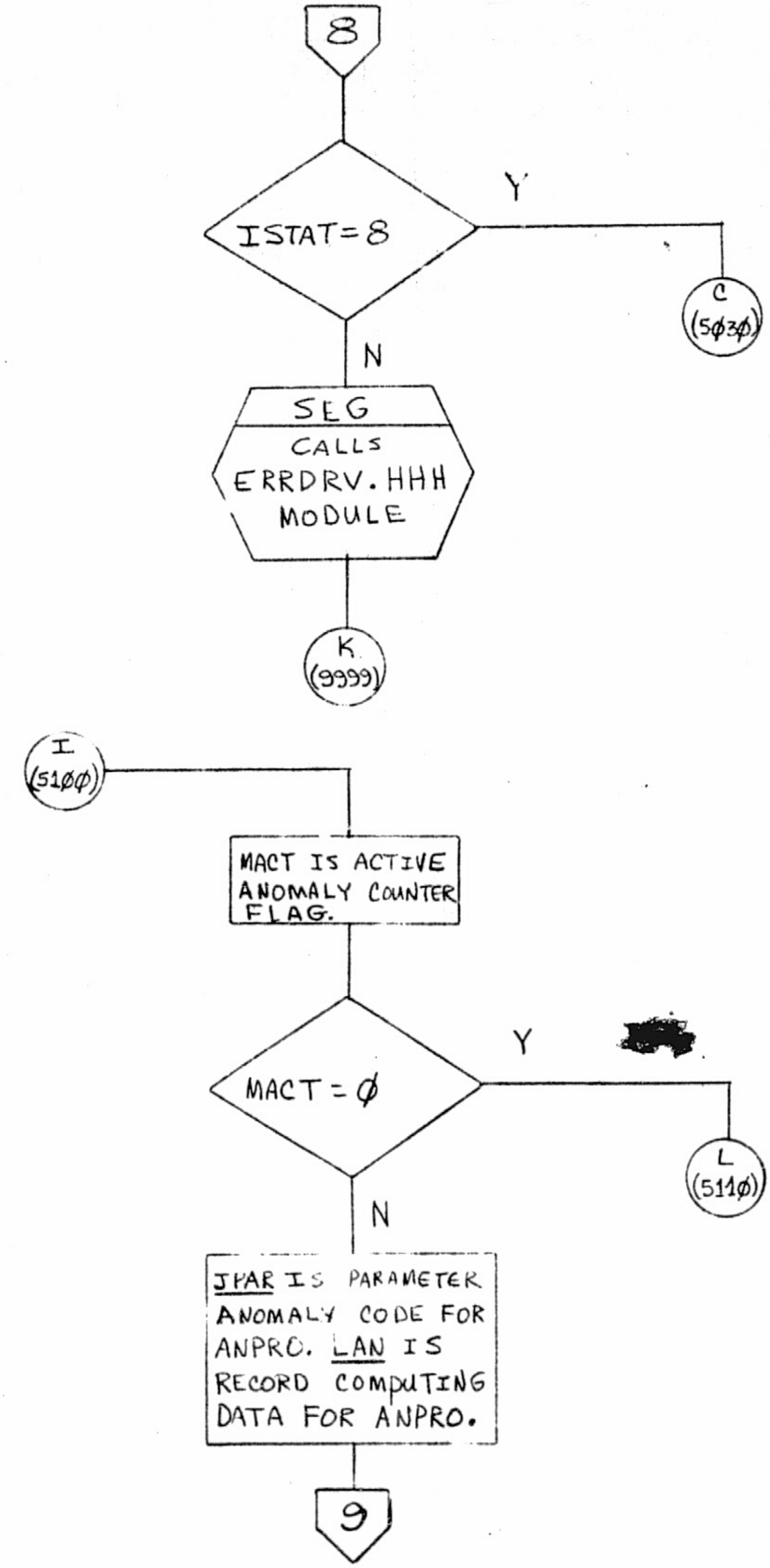

3
IS BIAS VOLTAGE HEADER WRITTEN ON SCRATCH TAPE
N
Y
IS BIAS VOLTAGE DATA WRITTEN ON SCRATCH TAPE
N
Y
IS DATA IN BIAS VOLTAGE TIME PERIOD
Y
N
CLEAR BIAS VOLTAGE DATA FLAGS:
IBVHDR = Ø
IBVFLG = Ø
IS CAL PERIOD HEADER WRITTEN ON SCRATCH TAPE
N
Y
D (5Ø7Ø)
4

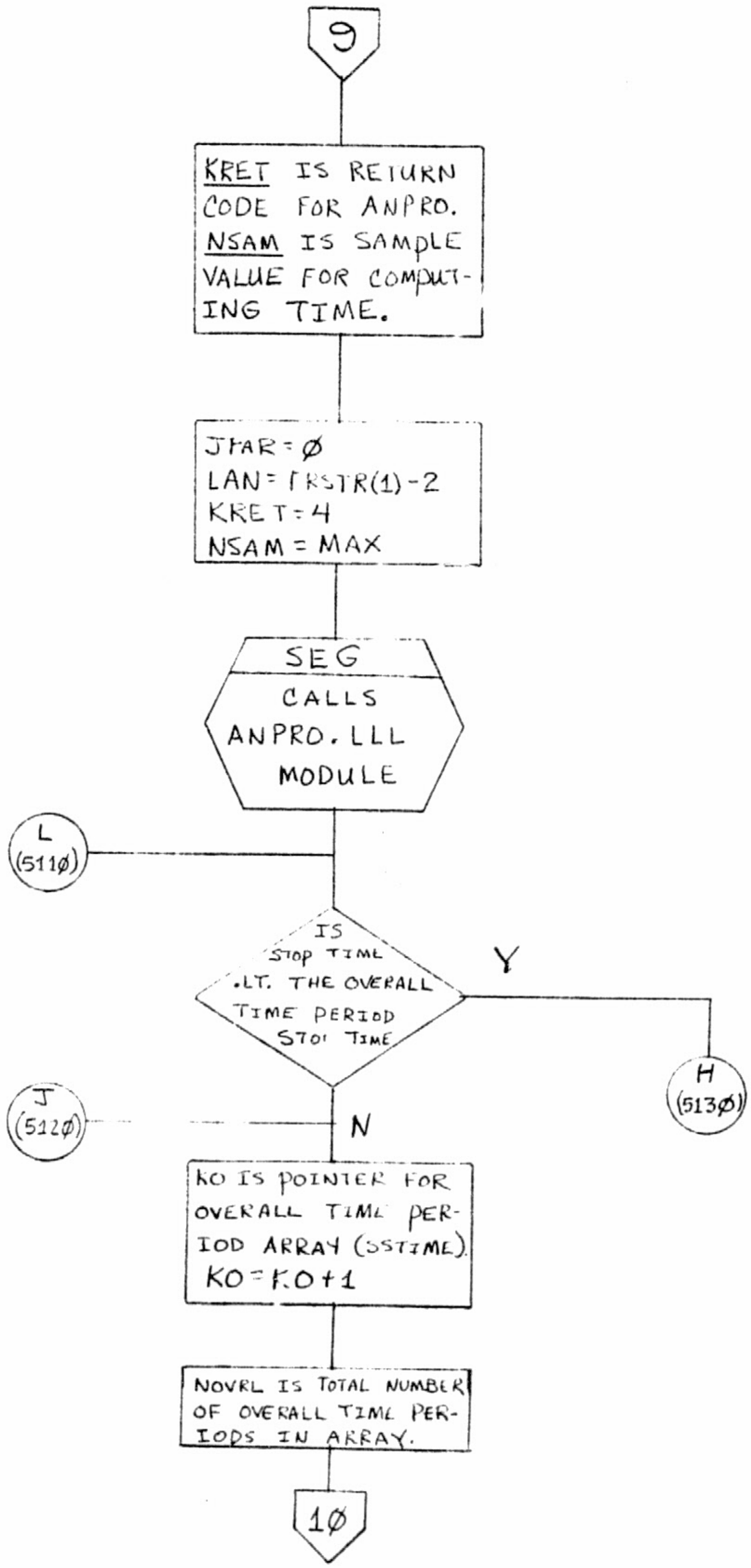

4
IS ALL OF CAL PERIOD DATA WRITTEN ON SCRATCH TAPE
N
D (5Ø7Ø)
Y
IS DATA WITHIN CAL PERIOD
Y
D (5Ø7Ø)
N
CLEAR CAL PERIOD FILE FLAGS:
ICLHDR=Ø
IDONE=Ø
IRSFLG=Ø
IRMFLG=Ø
D (5Ø7Ø)
SET MAXIMUM NUMBER OF FRAMES (MAX) TO READ FROM INPUT TAPE:
MAX = 18
SEG
CALLS DCMDRV.III MODULE
5

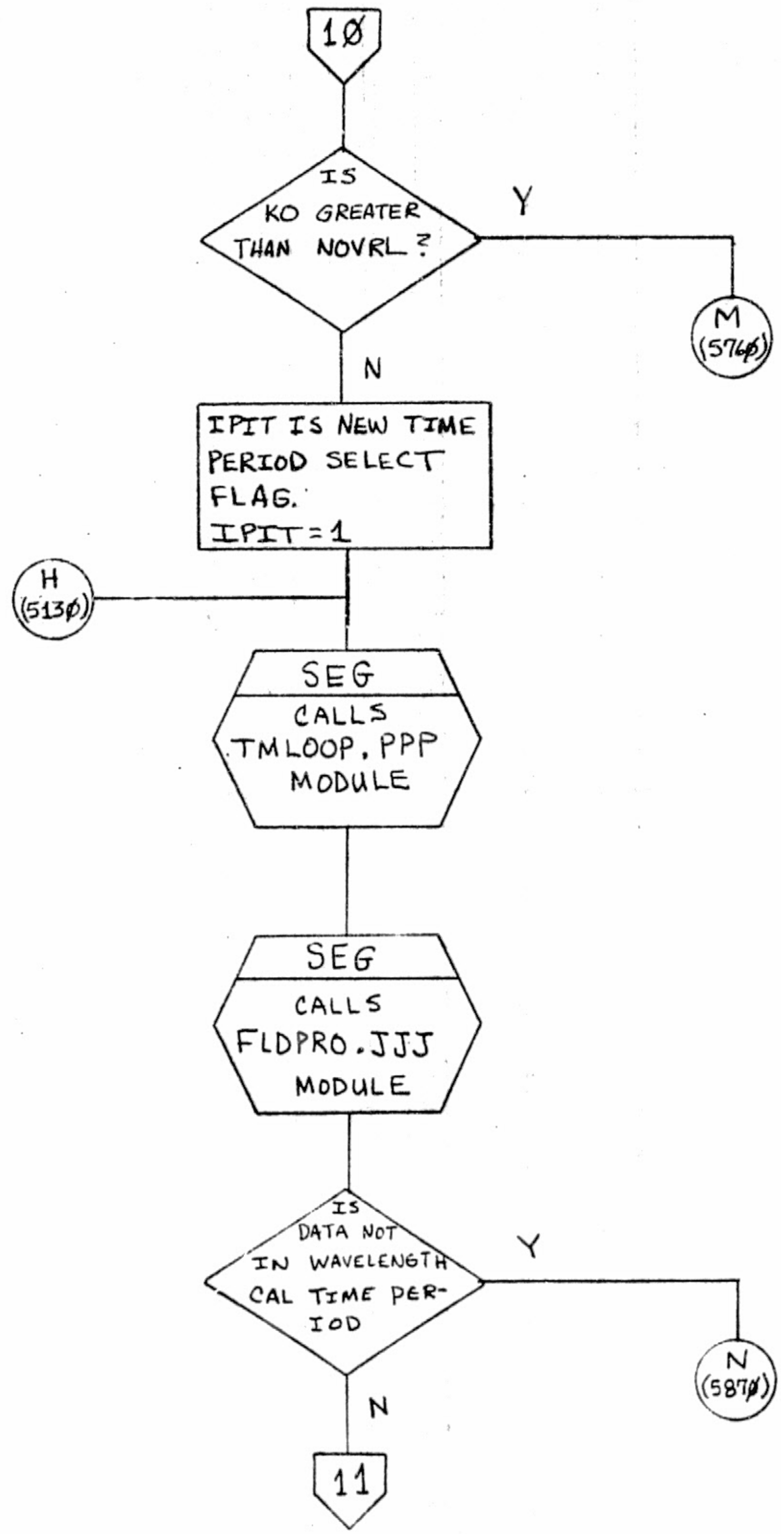

5
DEBUG PRINT
N
Y
E (5Ø71)
DEBUG PDUMPS TIMES (START ¦ STOP), STATUS (ISTAT), AND MAXIMUM NUMBER OF FRAMES READ (MAX).
E (5Ø71)
HAS FIRST FRAME TIME FOR QASUM BEEN DERIVED
Y
F (5Ø75)
N
START DERIVATION OF FIRST FRAME TIME (STTME).
AMOV
MOVE FRAME TIME (MEAS(69)) TO D.P. STTME.
AMOV
MOVE DELTA TIME (MEAS(85)) TO FRSPRC.
6

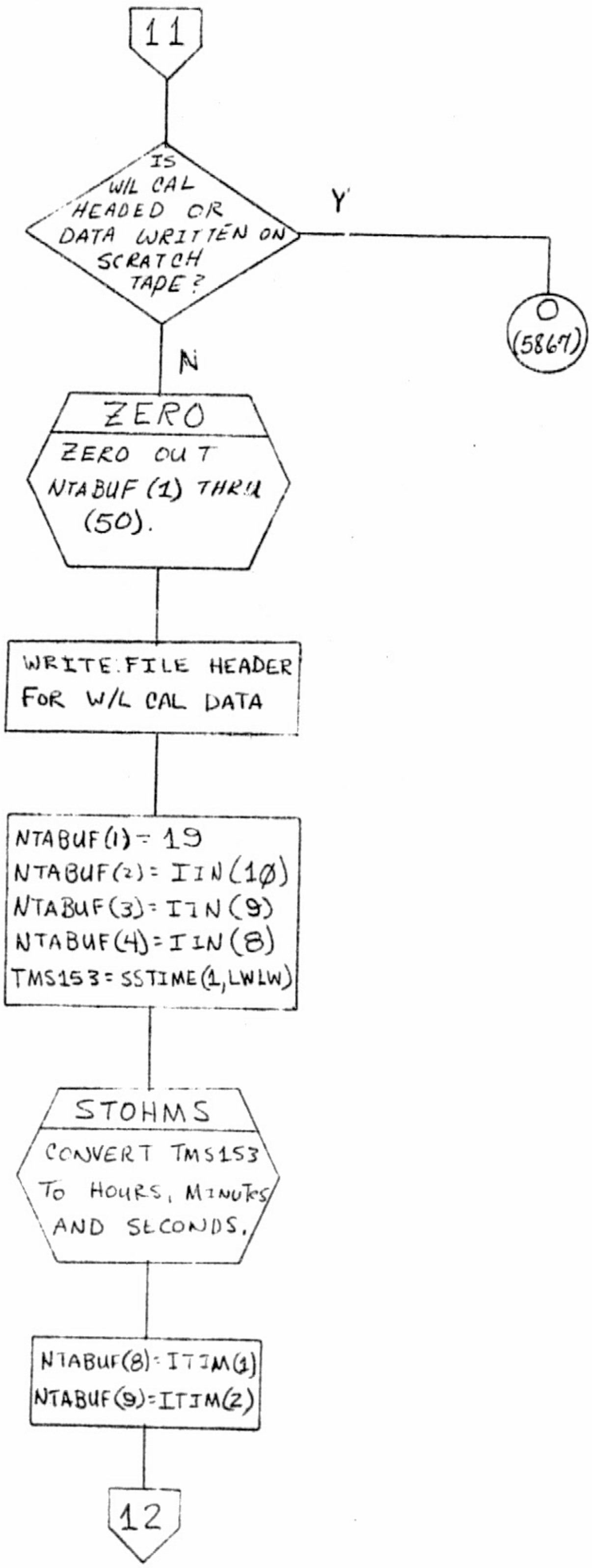

6
COMPUTE TOTAL DELTA TIME:
RNN=MAX/6
ISE=RNN
STTME= STTME +
(FRSPRC * ISE)
SET FIRST FRAME TIME FLAG (MFLG):
MFLG=1
F
(5Ø75)
ZERO FRAMES READ FROM INPUT TAPE
Y
G
(575Ø)
N
DCOMZN RETURN STATUS CODES: (ISTAT)
Ø= NORMAL
1= TIME OUT OF TOLERANCE
2= END OF TIME BEFORE MAX FRAMES RETURNED
6= BAD FRAME TIME
7= ALL FRAME TIMES BAD.
8=ALL TIMES BAD. NEXT TIMEL IS GOOD.
7

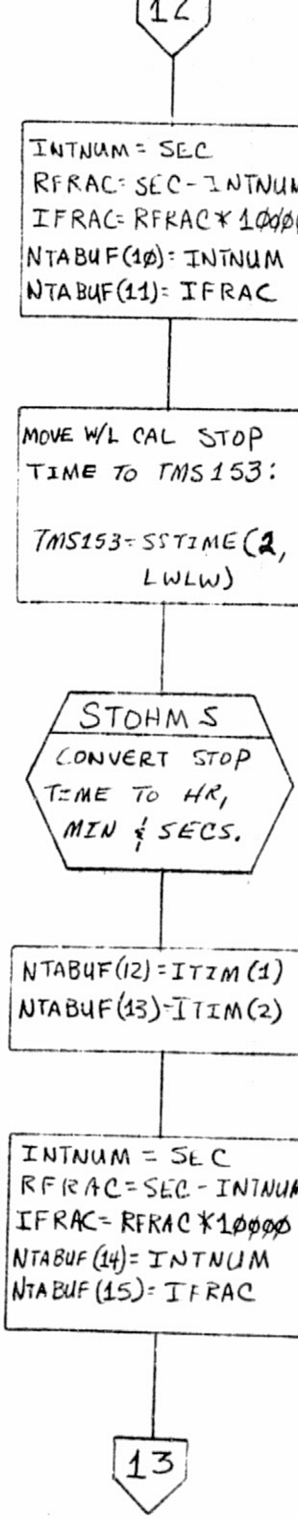

7
ISTAT = Ø
Y
H
(513Ø)
N
ISTAT = 1
Y
I
(51ØØ)
N
ISTAT = 2
Y
J
(512Ø)
N
ISTAT = 6
Y
C
(5Ø3Ø)
N
ISTAT = 7
Y
C
(5Ø3Ø)
8

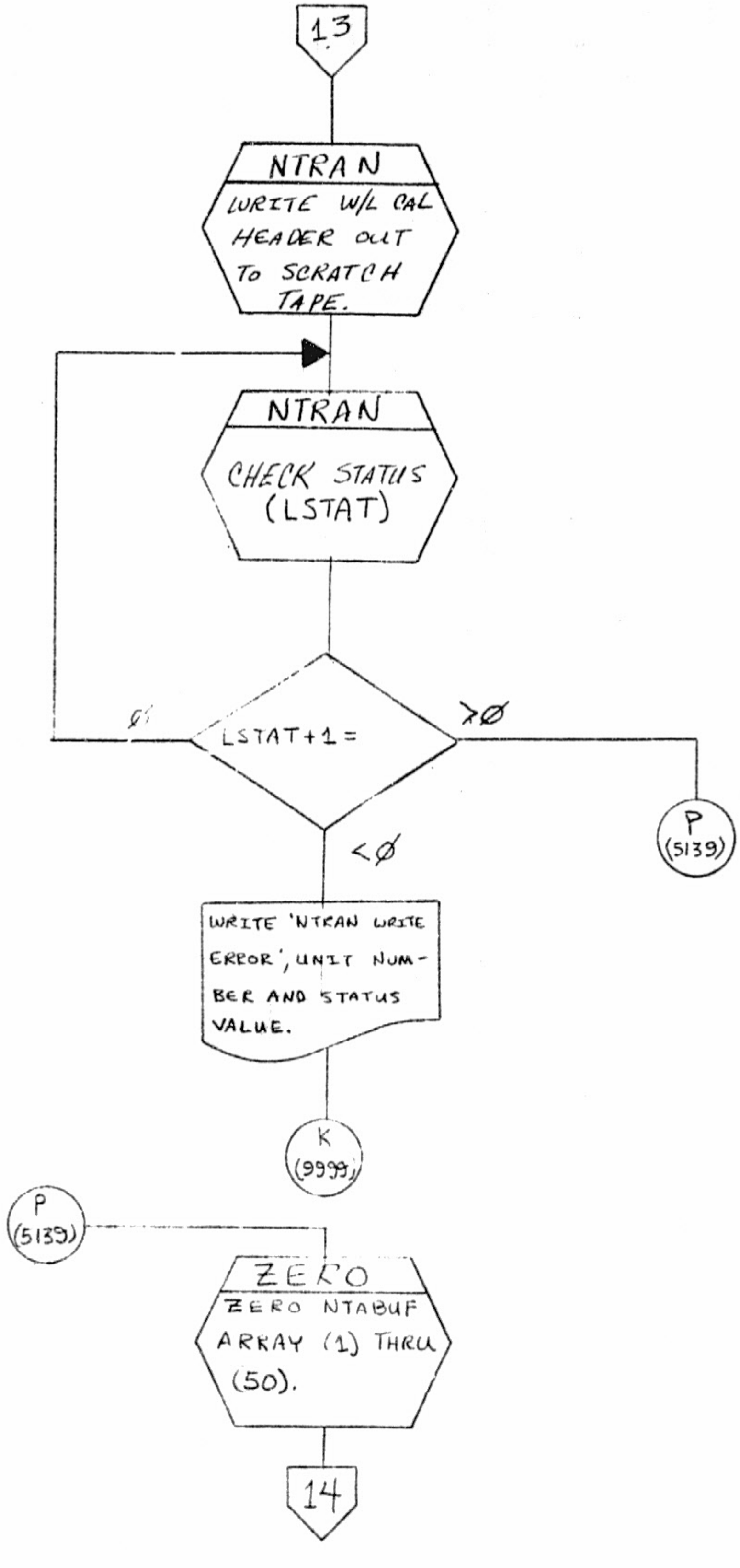

8
ISTAT=8
Y
C
(5Ø3Ø)
N
SEG
CALLS
ERRDRV.HHH
MODULE
K
(9999)
I
(51ØØ)
MACT IS ACTIVE
ANOMALY COUNTER
FLAG.
MACT = Ø
Y
L
(511Ø)
N
JPAR IS PARAMETER
ANOMALY CODE FOR
ANPRO. LAN IS
RECORD COMPUTING
DATA FOR ANPRO.
9

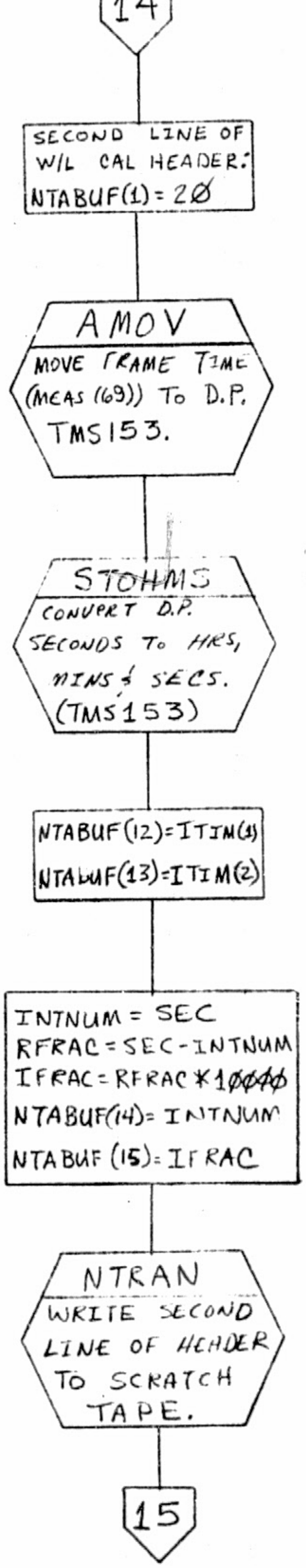

9
KRET IS RETURN CODE FOR ANPRO.
NSAM IS SAMPLE VALUE FOR COMPUTING TIME.
JPAR = Ø
LAN = FRSTR(1) - 2
KRET = 4
NSAM = MAX
SEG
CALLS ANPRO.LLL MODULE
L (511Ø)
IS STOP TIME .LT. THE OVERALL TIME PERIOD STOP TIME
Y
H (513Ø)
J (512Ø)
N
KO IS POINTER FOR OVERALL TIME PERIOD ARRAY (SSTIME).
KO = KO + 1
NOVRL IS TOTAL NUMBER OF OVERALL TIME PERIODS IN ARRAY.
1Ø

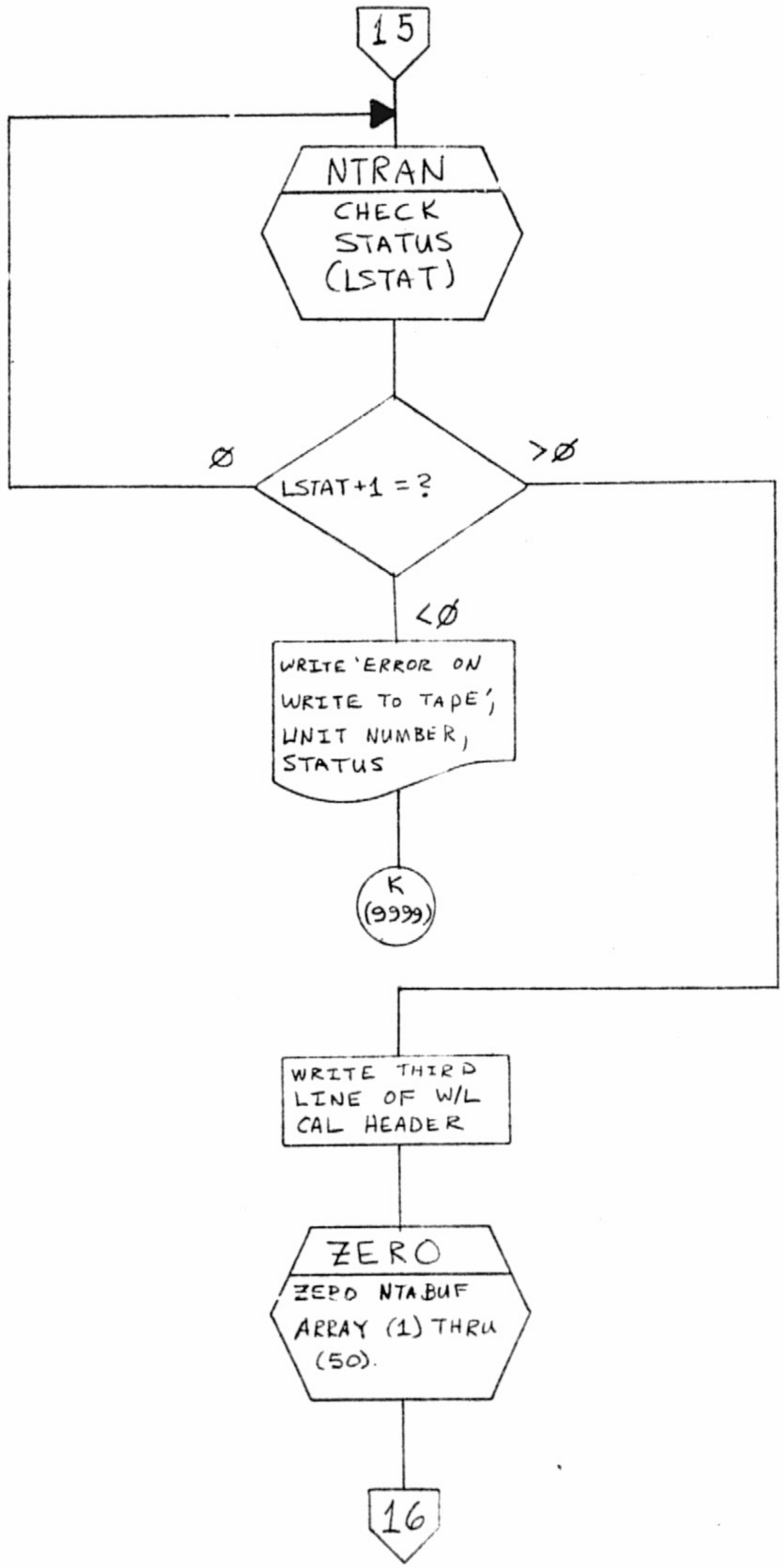

10
IS KO GREATER THAN NOVRL ?
Y
M (576∅)
N
IPIT IS NEW TIME PERIOD SELECT FLAG.
IPIT = 1
H (513∅)
SEG
CALLS TMLOOP. PPP MODULE
SEG
CALLS FLDPRO.JJJ MODULE
IS DATA NOT IN WAVELENGTH CAL TIME PERIOD
Y
N (587∅)
N
11

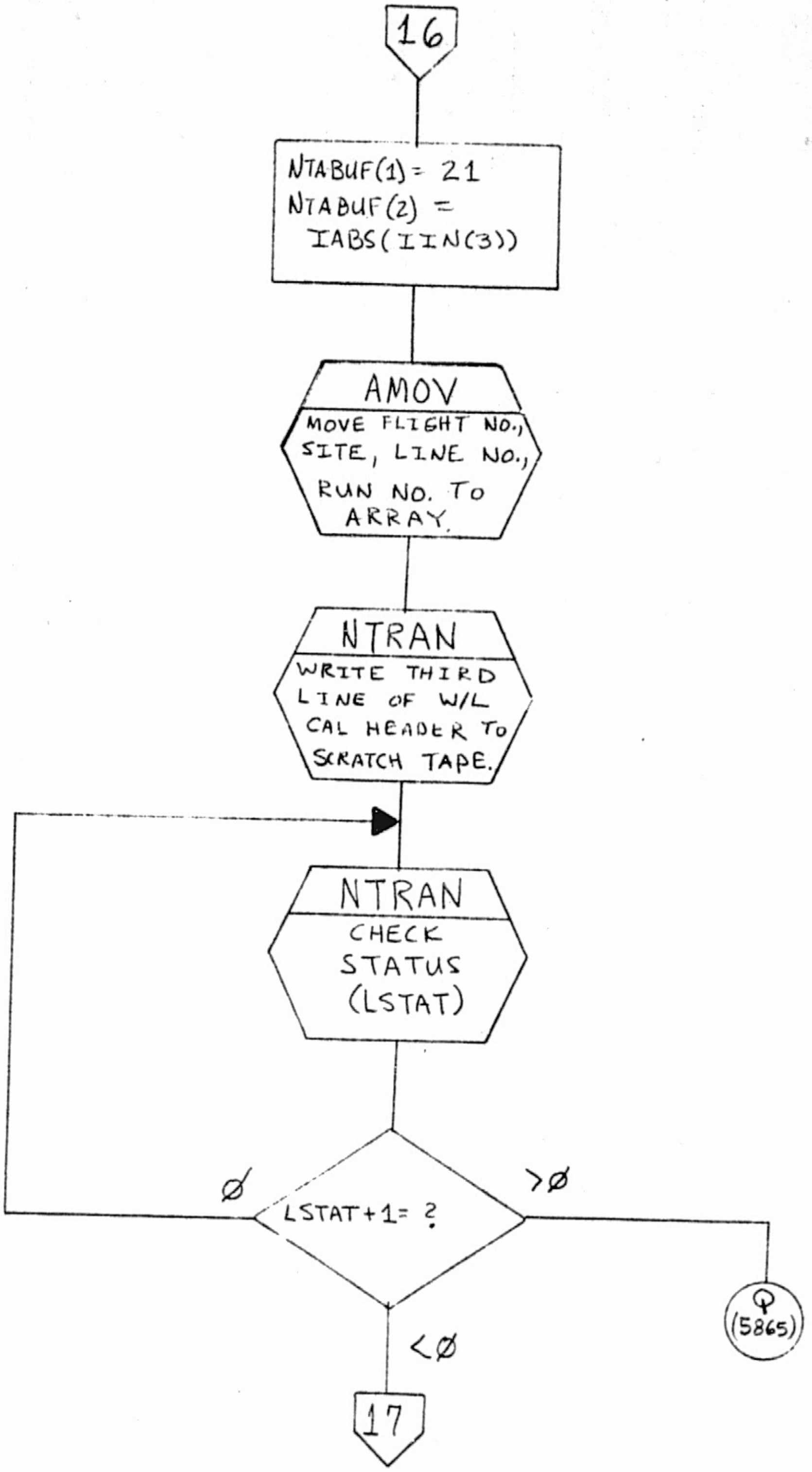

11
IS W/L CAL HEADED OR DATA WRITTEN ON SCRATCH TAPE?
Y
O
(5867)
N
ZERO
ZERO OUT NTABUF (1) THRU (50).
WRITE FILE HEADER FOR W/L CAL DATA
NTABUF(1) = 19
NTABUF(2) = IIN(1∅)
NTABUF(3) = IIN(9)
NTABUF(4) = IIN(8)
TMS153 = SSTIME(1,LWLW)
STOHMS
CONVERT TMS153 TO HOURS, MINUTES AND SECONDS.
NTABUF(8) = ITIM(1)
NTABUF(9) = ITIM(2)
12

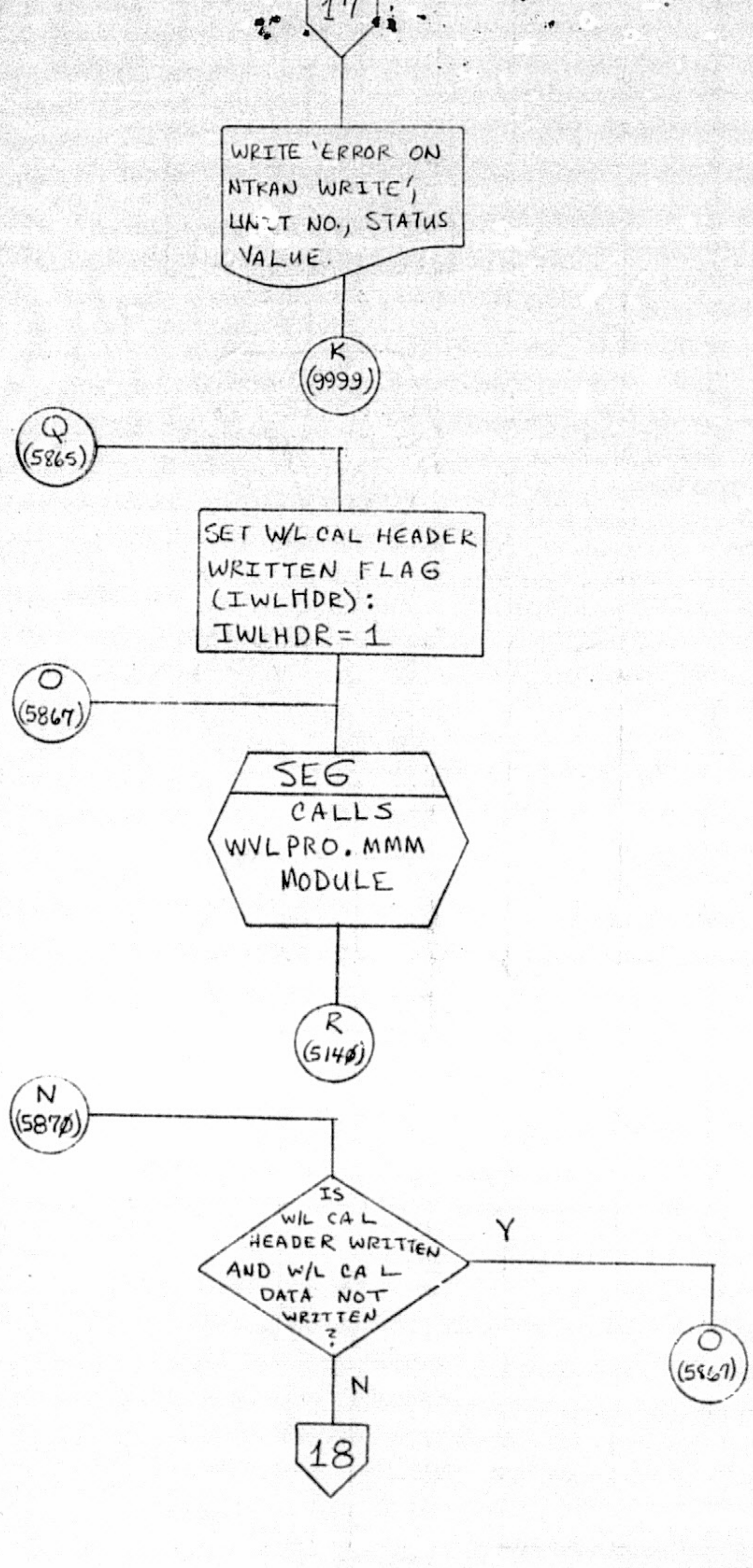

12
INTNUM = SEC
RFRAC = SEC - INTNUM
IFRAC = RFRAC * 10000
NTABUF(10) = INTNUM
NTABUF(11) = IFRAC
MOVE W/L CAL STOP TIME TO TMS153:
TMS153 = SSTIME(2, LWLW)
STOHMS
CONVERT STOP TIME TO HR, MIN & SECS.
NTABUF(12) = ITIM(1)
NTABUF(13) = ITIM(2)
INTNUM = SEC
RFRAC = SEC - INTNUM
IFRAC = RFRAC * 10000
NTABUF(14) = INTNUM
NTABUF(15) = IFRAC
13

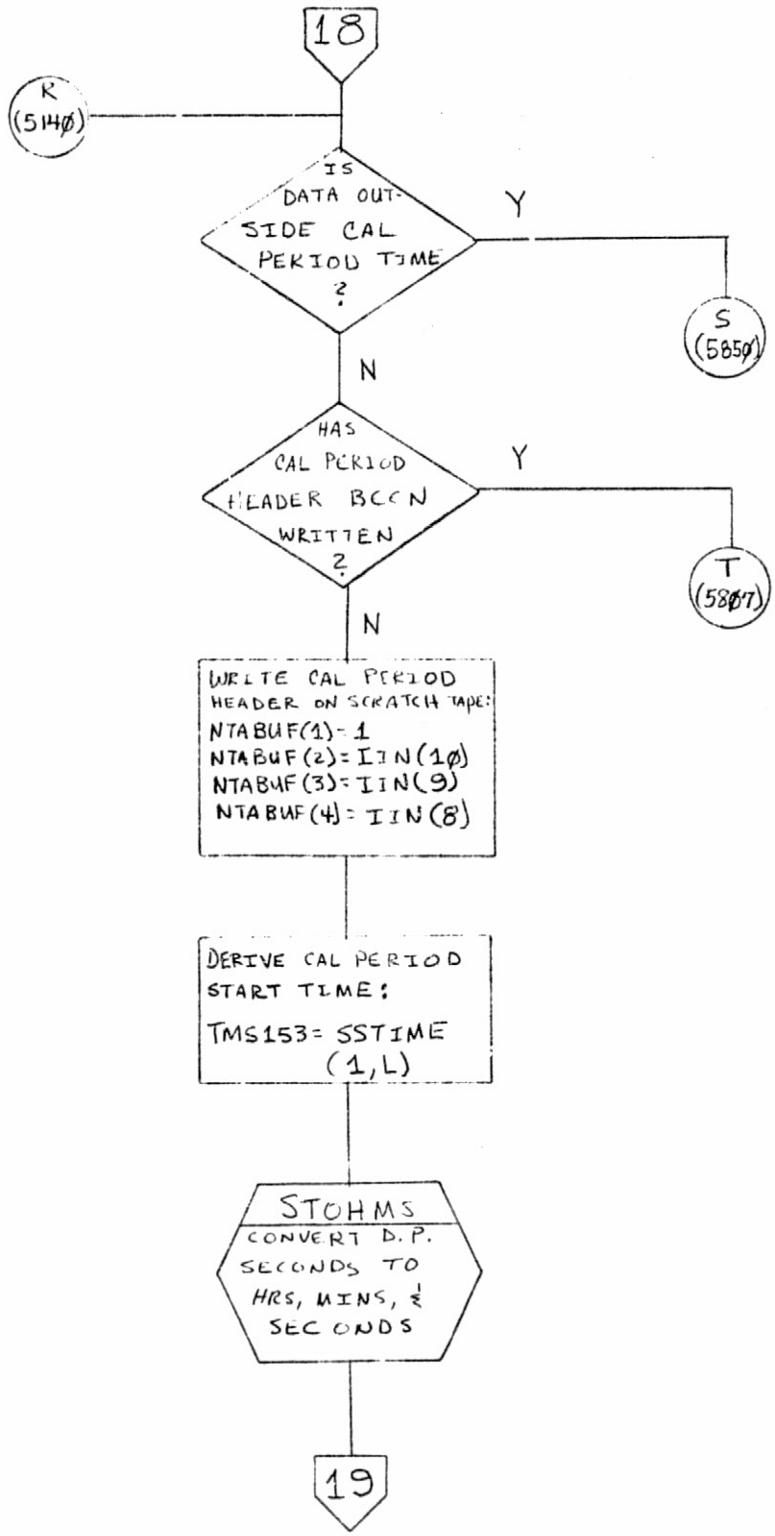

13
NTRAN
WRITE W/L CAL HEADER OUT TO SCRATCH TAPE.
NTRAN
CHECK STATUS (LSTAT)
LSTAT+1 =
Ø
>Ø
<Ø
P (5139)
WRITE 'NTRAN WRITE ERROR', UNIT NUMBER AND STATUS VALUE.
K (9999)
P (5139)
ZERO
ZERO NTABUF ARRAY (1) THRU (50).
14

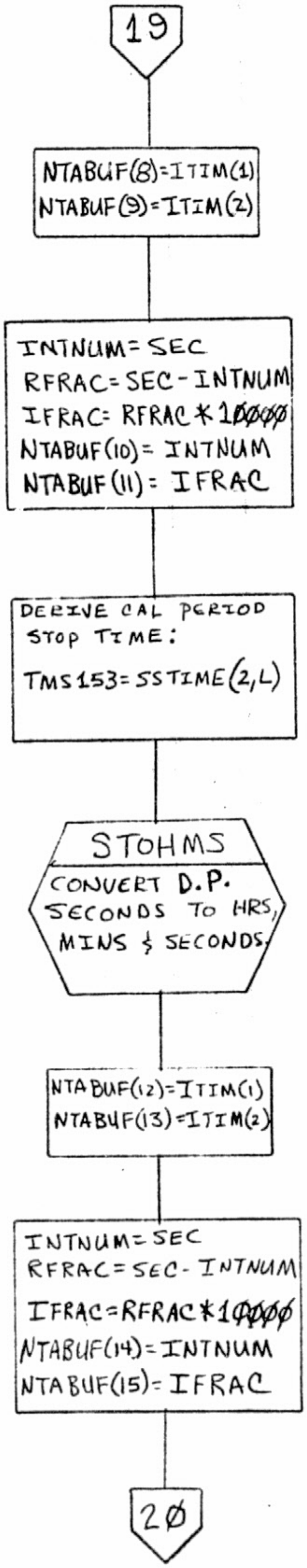

14
SECOND LINE OF W/L CAL HEADER:
NTABUF(1) = 2Ø
AMOV
MOVE FRAME TIME (MEAS (69)) TO D.P. TMS153.
STOHMS
CONVERT D.P. SECONDS TO HRS, MINS & SECS. (TMS153)
NTABUF(12)=ITIM(1)
NTABUF(13)=ITIM(2)
INTNUM = SEC
RFRAC = SEC-INTNUM
IFRAC = RFRAC * 1ØØØØ
NTABUF(14)= INTNUM
NTABUF (15)= IFRAC
NTRAN
WRITE SECOND LINE OF HEADER TO SCRATCH TAPE.
15

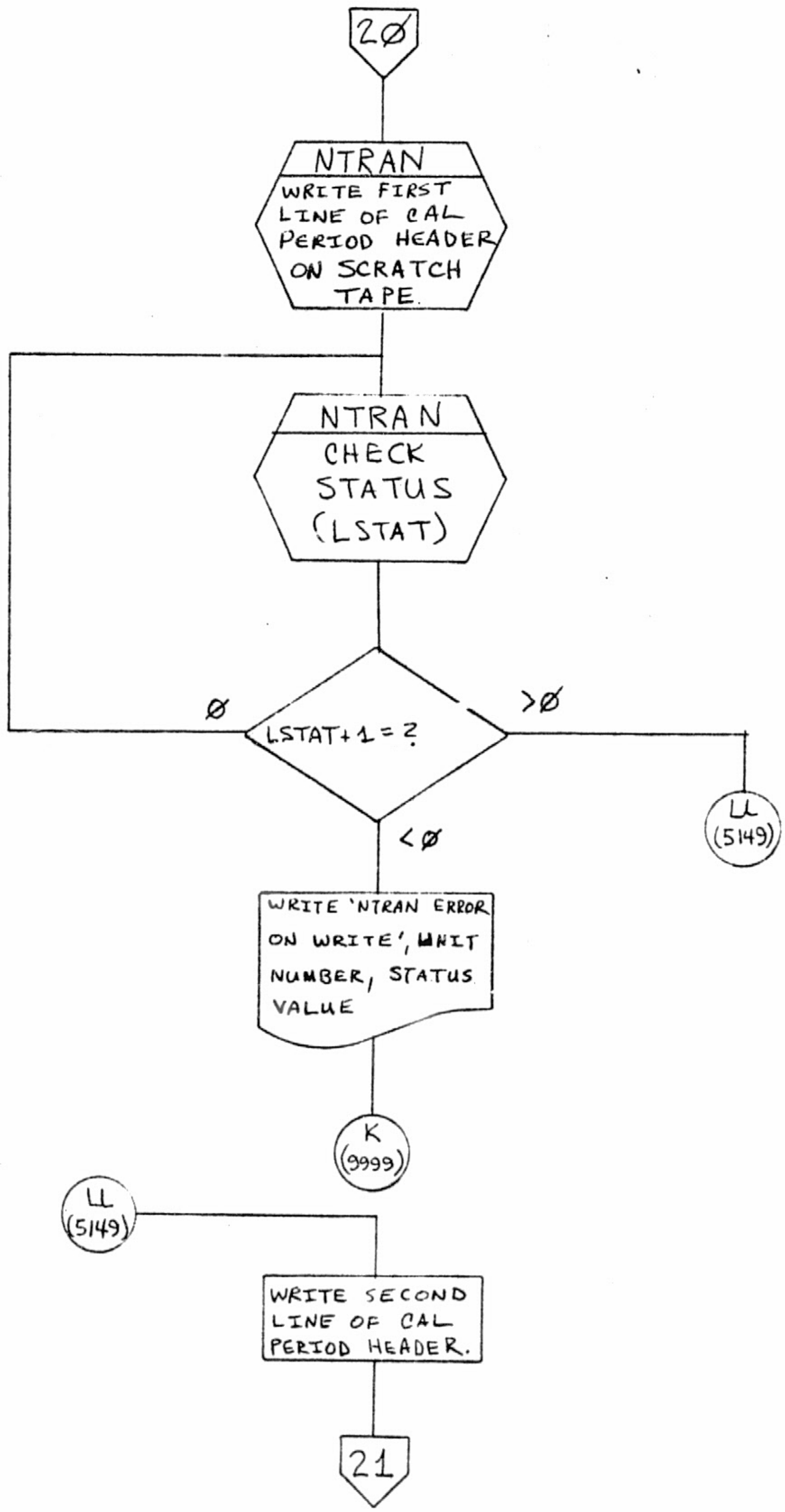

15
NTRAN
CHECK STATUS (LSTAT)
Ø
>Ø
LSTAT+1 =?
<Ø
WRITE 'ERROR ON WRITE TO TAPE', UNIT NUMBER, STATUS
K (9999)
WRITE THIRD LINE OF W/L CAL HEADER
ZERO
ZERO NTABUF ARRAY (1) THRU (50).
16

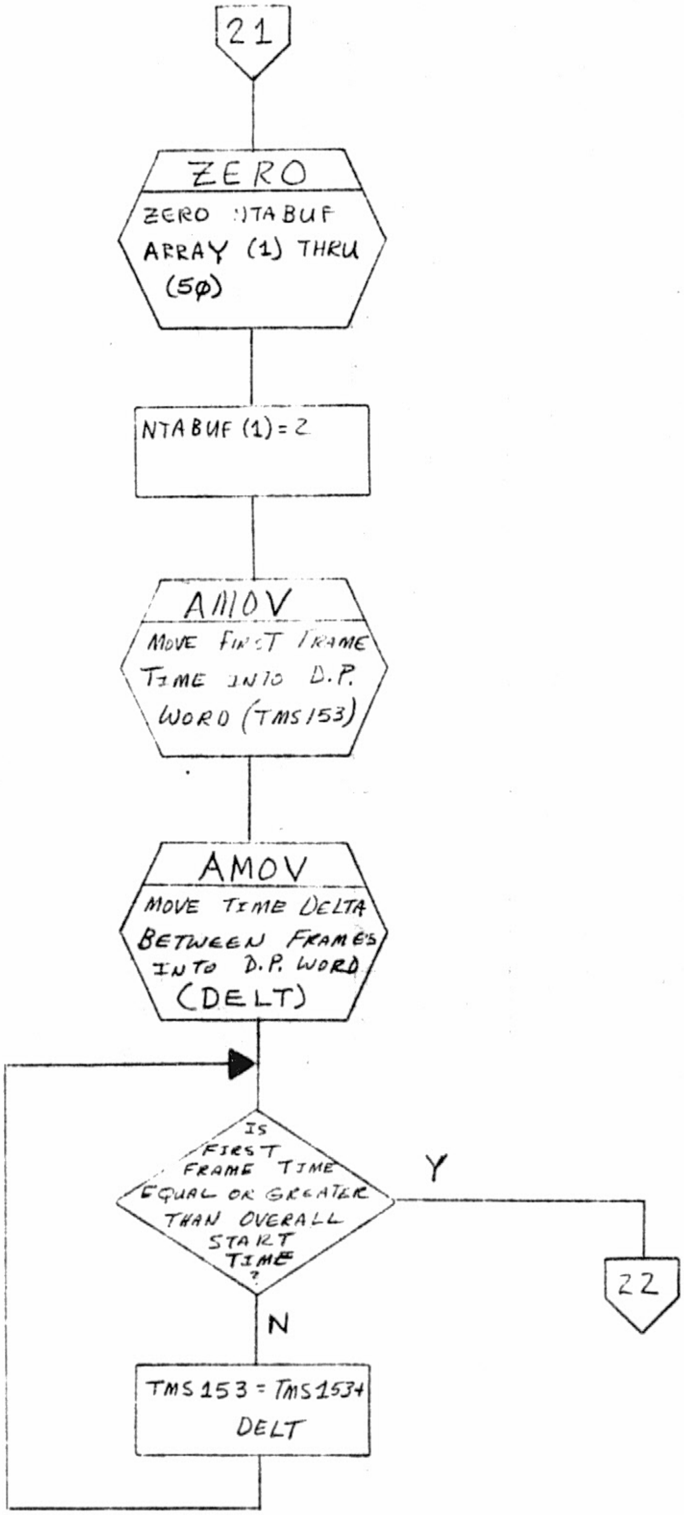

16
NTABUF(1) = 21
NTABUF(2) =
IABS(IIN(3))
AMOV
MOVE FLIGHT NO., SITE, LINE NO., RUN NO. TO ARRAY.
NTRAN
WRITE THIRD LINE OF W/L CAL HEADER TO SCRATCH TAPE.
NTRAN
CHECK STATUS (LSTAT)
LSTAT+1 = ?
Ø
>Ø
<Ø
(5865)
17

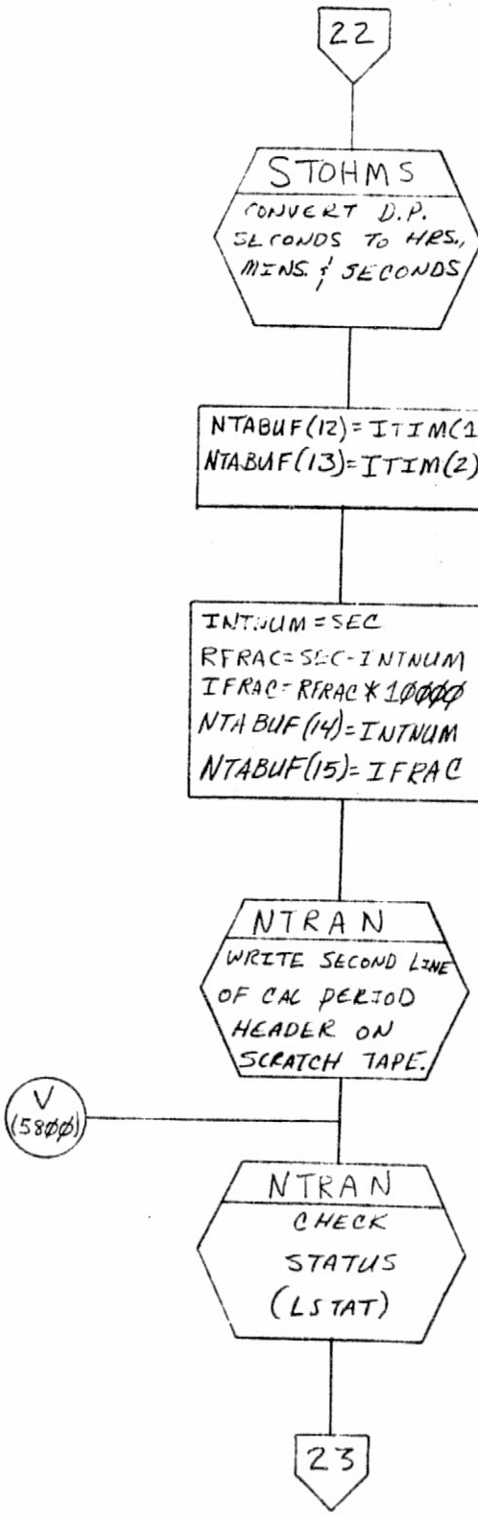

17
WRITE 'ERROR ON NTRAN WRITE', UNIT NO., STATUS VALUE.
K (9999)
Q (5865)
SET W/L CAL HEADER WRITTEN FLAG (IWLHDR): IWLHDR = 1
O (5867)
SEG
CALLS WVLPRO.MMM MODULE
R (5140)
N (5870)
IS W/L CAL HEADER WRITTEN AND W/L CAL DATA NOT WRITTEN ?
Y
O (5867)
N
18

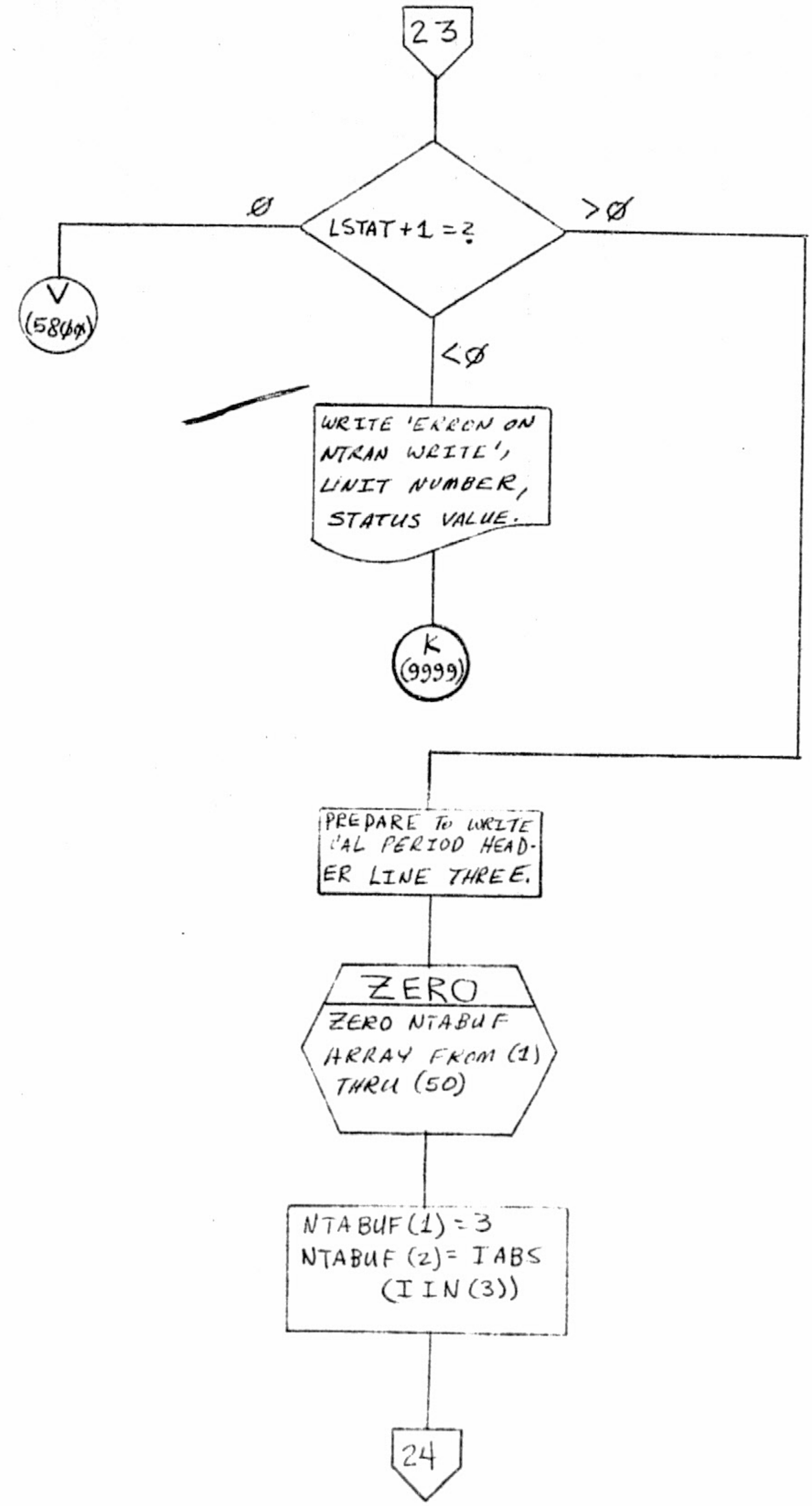

18
R
(514Ø)
IS DATA OUTSIDE CAL PERIOD TIME ?
Y
S
(585Ø)
N
HAS CAL PERIOD HEADER BEEN WRITTEN ?
Y
T
(58Ø7)
N
WRITE CAL PERIOD HEADER ON SCRATCH TAPE:
NTABUF(1)=1
NTABUF(2)=IIN(1Ø)
NTABUF(3)=IIN(9)
NTABUF(4)=IIN(8)
DERIVE CAL PERIOD START TIME:
TMS153=SSTIME (1,L)
STOHMS
CONVERT D.P. SECONDS TO HRS, MINS, & SECONDS
19

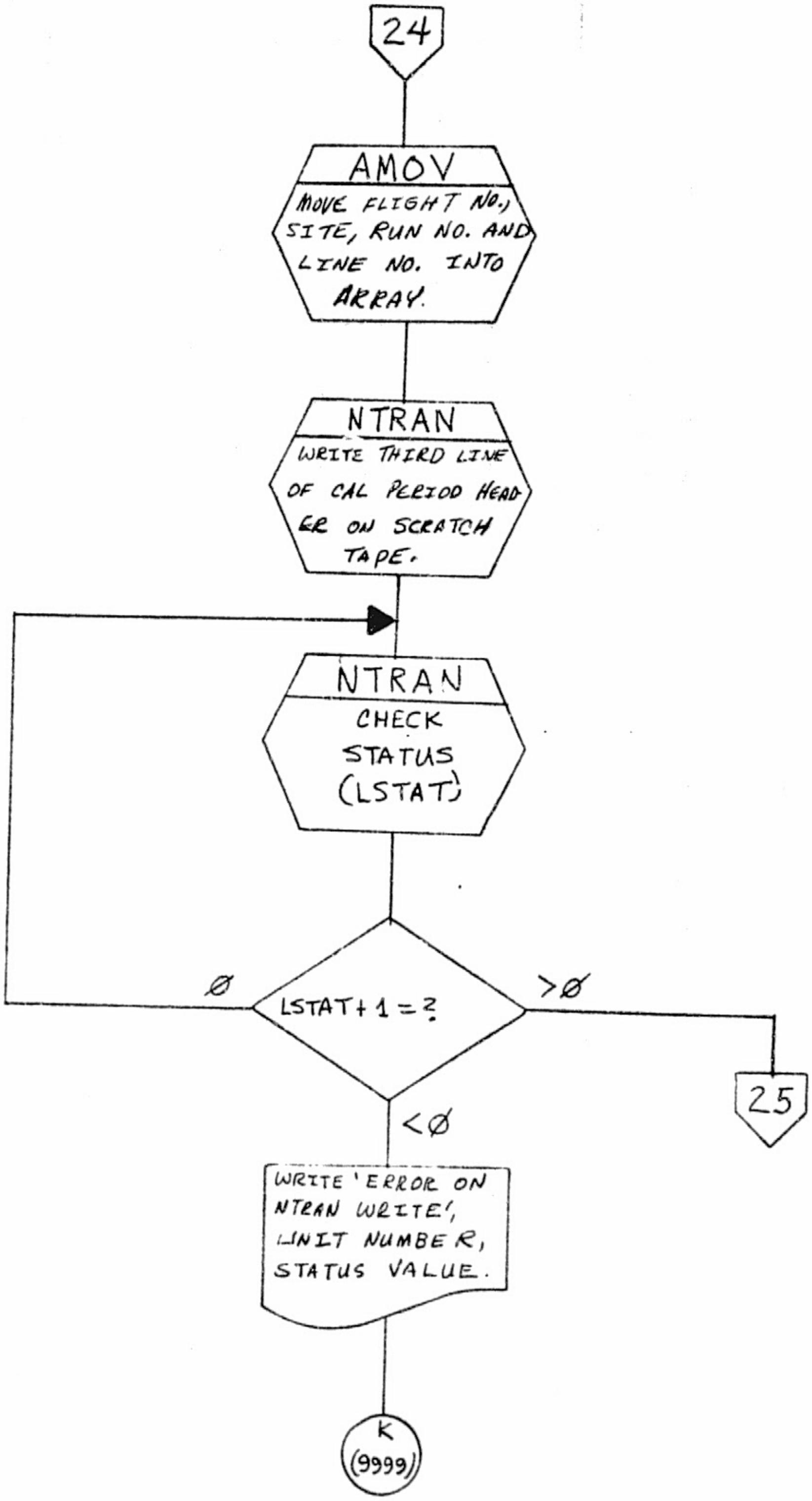

19
NTABUF(8)=ITIM(1)
NTABUF(9)=ITIM(2)
INTNUM=SEC
RFRAC=SEC-INTNUM
IFRAC=RFRAC*1ØØØØ
NTABUF(10)=INTNUM
NTABUF(11)=IFRAC
DERIVE CAL PERIOD STOP TIME:
TMS153=SSTIME(2,L)
STOHMS
CONVERT D.P. SECONDS TO HRS, MINS & SECONDS
NTABUF(12)=ITIM(1)
NTABUF(13)=ITIM(2)
INTNUM=SEC
RFRAC=SEC-INTNUM
IFRAC=RFRAC*1ØØØØ
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
2Ø


20
NTRAN
WRITE FIRST LINE OF CAL PERIOD HEADER ON SCRATCH TAPE.
NTRAN
CHECK STATUS (LSTAT)
LSTAT+1 = ?
Ø
>Ø
<Ø
LL (5149)
WRITE 'NTRAN ERROR ON WRITE', UNIT NUMBER, STATUS VALUE
K (9999)
LL (5149)
WRITE SECOND LINE OF CAL PERIOD HEADER.
21

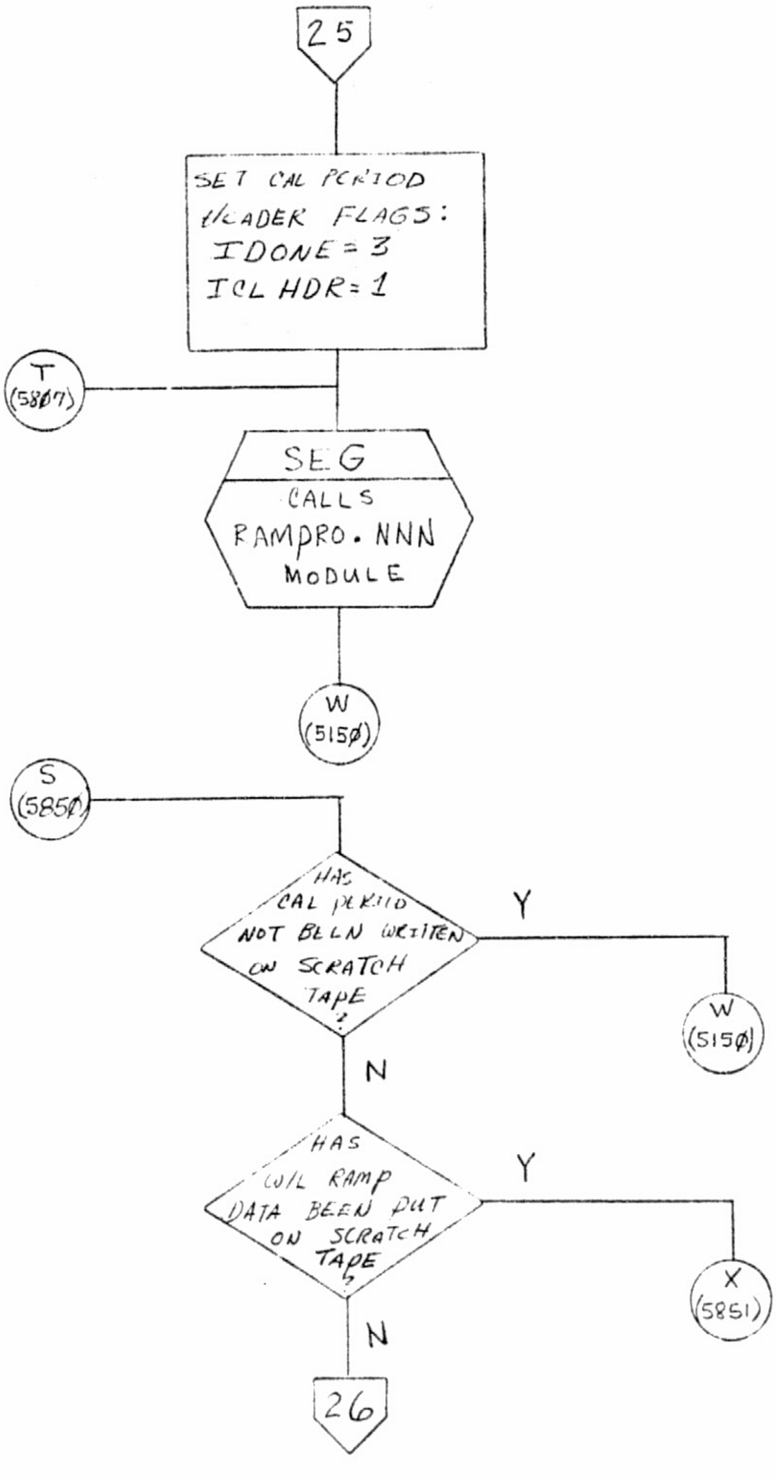

21
ZERO
ZERO NTABUF ARRAY (1) THRU (5φ)
NTABUF (1) = 2
AMOV
MOVE FIRST FRAME TIME INTO D.P. WORD (TMS153)
AMOV
MOVE TIME DELTA BETWEEN FRAMES INTO D.P. WORD (DELT)
IS FIRST FRAME TIME EQUAL OR GREATER THAN OVERALL START TIME ?
Y
22
N
TMS153 = TMS153+ DELT

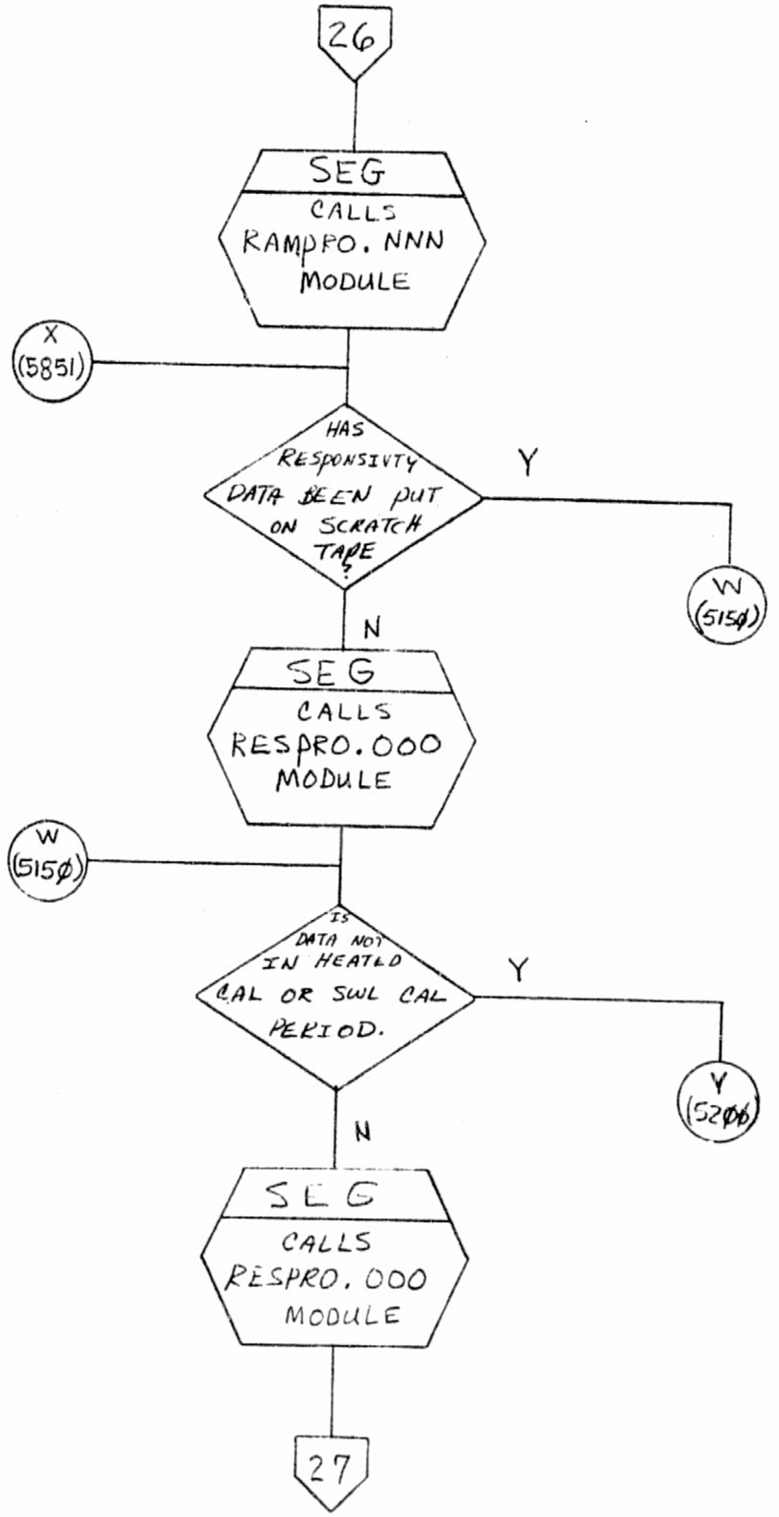

22
STOHMS
CONVERT D.P. SECONDS TO HRS., MINS. & SECONDS
NTABUF(12)=ITIM(1)
NTABUF(13)=ITIM(2)
INTNUM=SEC
RFRAC=SEC-INTNUM
IFRAC=RFRAC*10000
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
NTRAN
WRITE SECOND LINE OF CAL PERIOD HEADER ON SCRATCH TAPE.
V (5800)
NTRAN
CHECK STATUS (LSTAT)
23

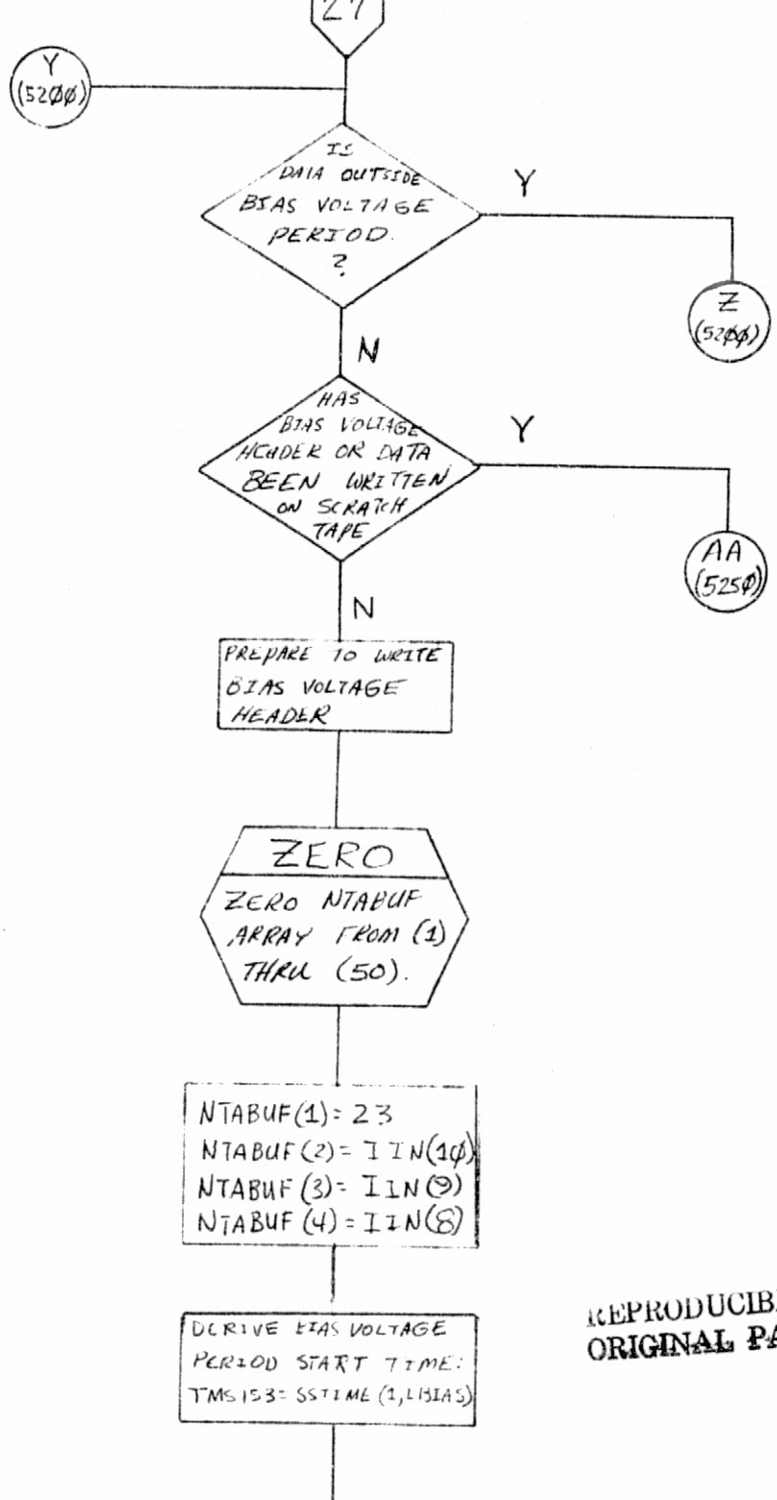

23
LSTAT+1 = ?
Ø
>Ø
<Ø
V
(58ØØ)
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE.
K
(9999)
PREPARE TO WRITE CAL PERIOD HEADER LINE THREE.
ZERO
ZERO NTABUF ARRAY FROM (1) THRU (50)
NTABUF(1) = 3
NTABUF(2) = IABS (IIN(3))
24

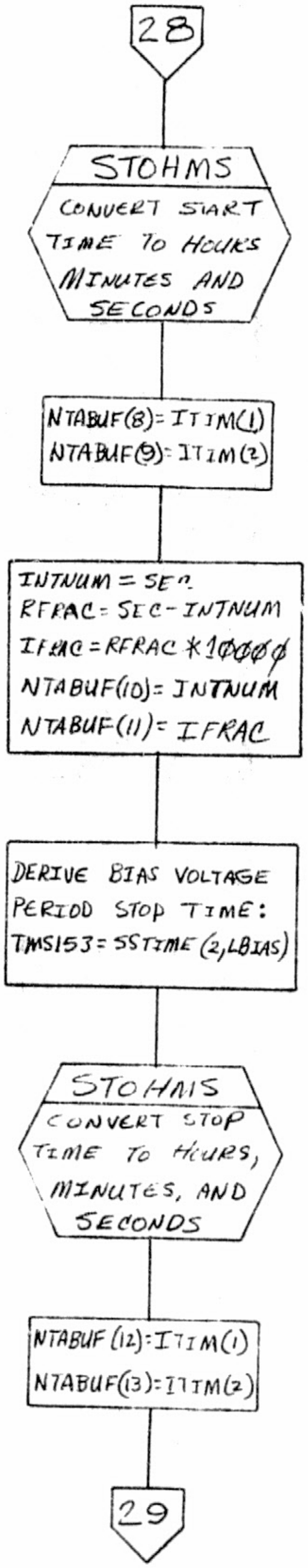


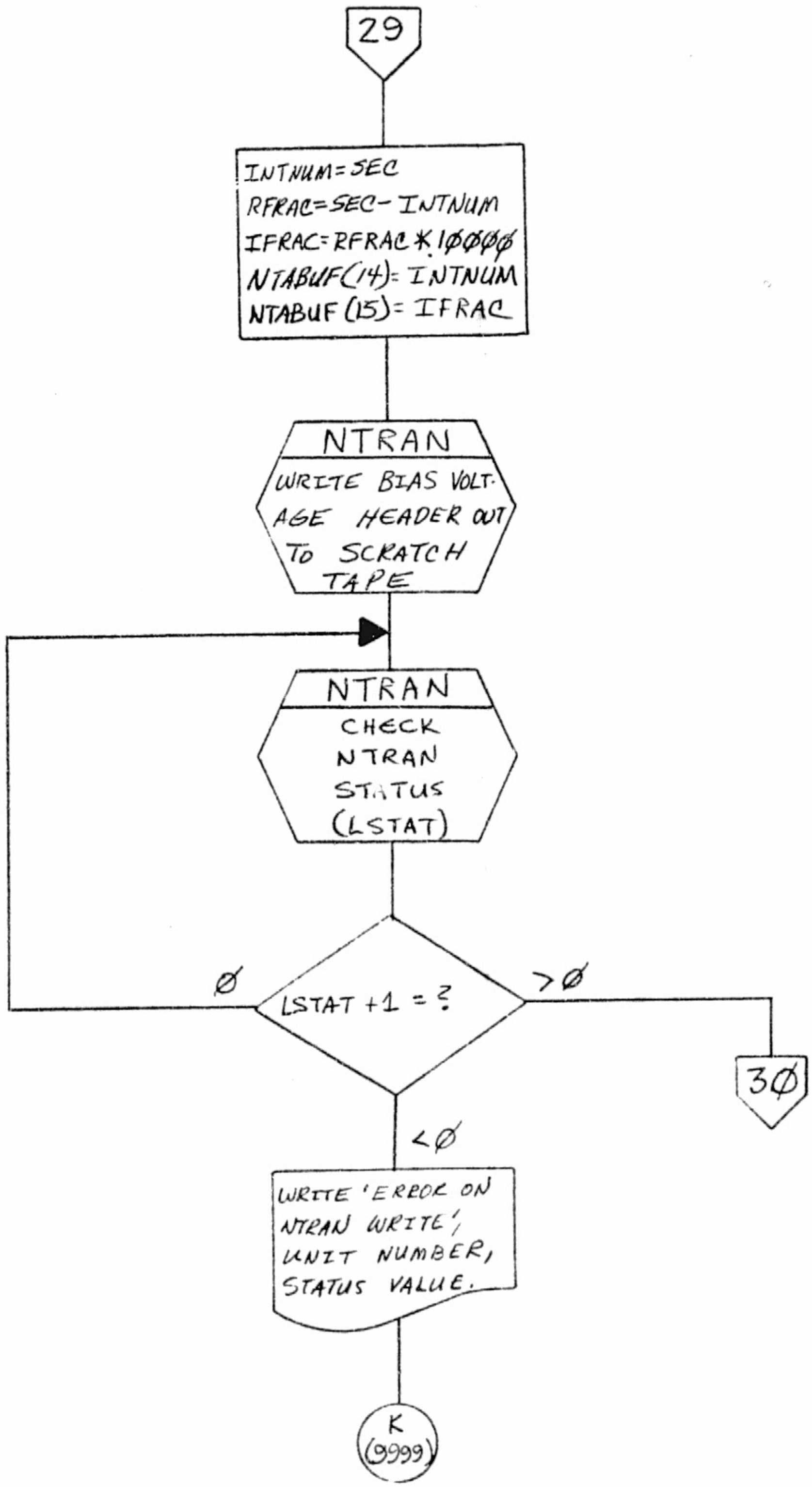
24
AMOV
MOVE FLIGHT NO., SITE, RUN NO. AND LINE NO. INTO ARRAY.
NTRAN
WRITE THIRD LINE OF CAL PERIOD HEADER ON SCRATCH TAPE.
NTRAN
CHECK STATUS (LSTAT)
Ø
LSTAT+1 = ?
>Ø
25
<Ø
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE.
K (9999)

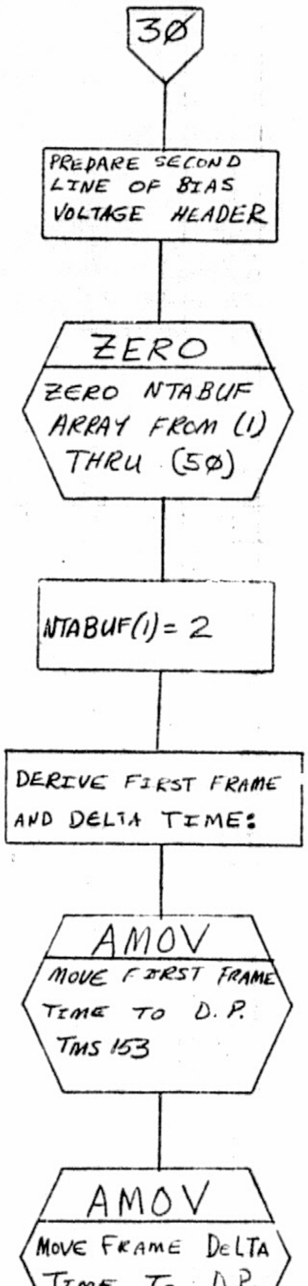

25
SET CAL PERIOD
HEADER FLAGS:
IDONE = 3
ICL HDR = 1
T
(58Ø7)
SEG G
CALLS
RAMPRO. NNN
MODULE
W
(515Ø)
S
(585Ø)
HAS
CAL PERIOD
NOT BEEN WRITTEN
ON SCRATCH
TAPE
?
Y
W
(515Ø)
N
HAS
W/L RAMP
DATA BEEN PUT
ON SCRATCH
TAPE
?
Y
X
(5851)
N
26

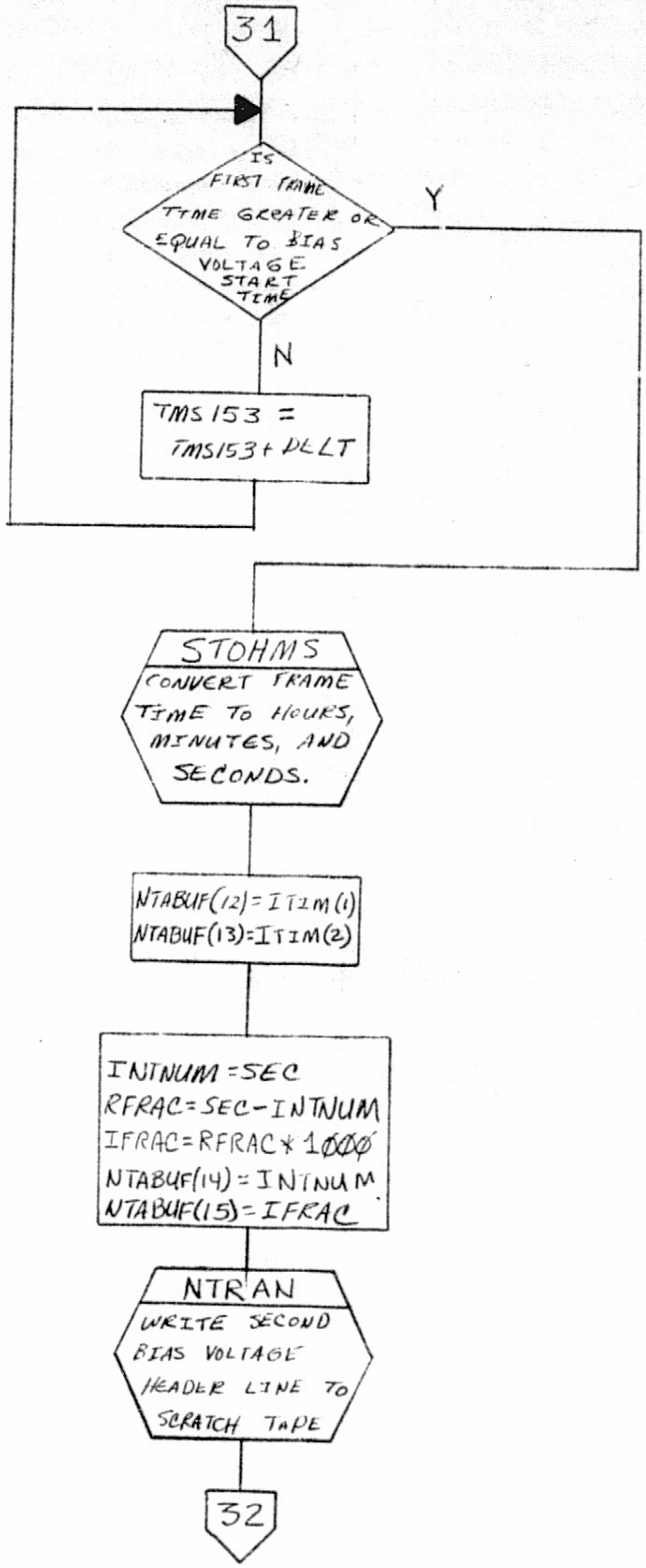

26
SEG
CALLS
RAMPRO. NNN
MODULE
X
(5851)
HAS
RESPONSIVTY
DATA BEEN PUT
ON SCRATCH
TAPE
?
Y
W
(515ϕ)
N
SEG
CALLS
RESPRO.000
MODULE
W
(515ϕ)
IS
DATA NOT
IN HEATED
CAL OR SWL CAL
PERIOD.
Y
Y
(52ϕϕ)
N
SEG
CALLS
RESPRO. 000
MODULE
27

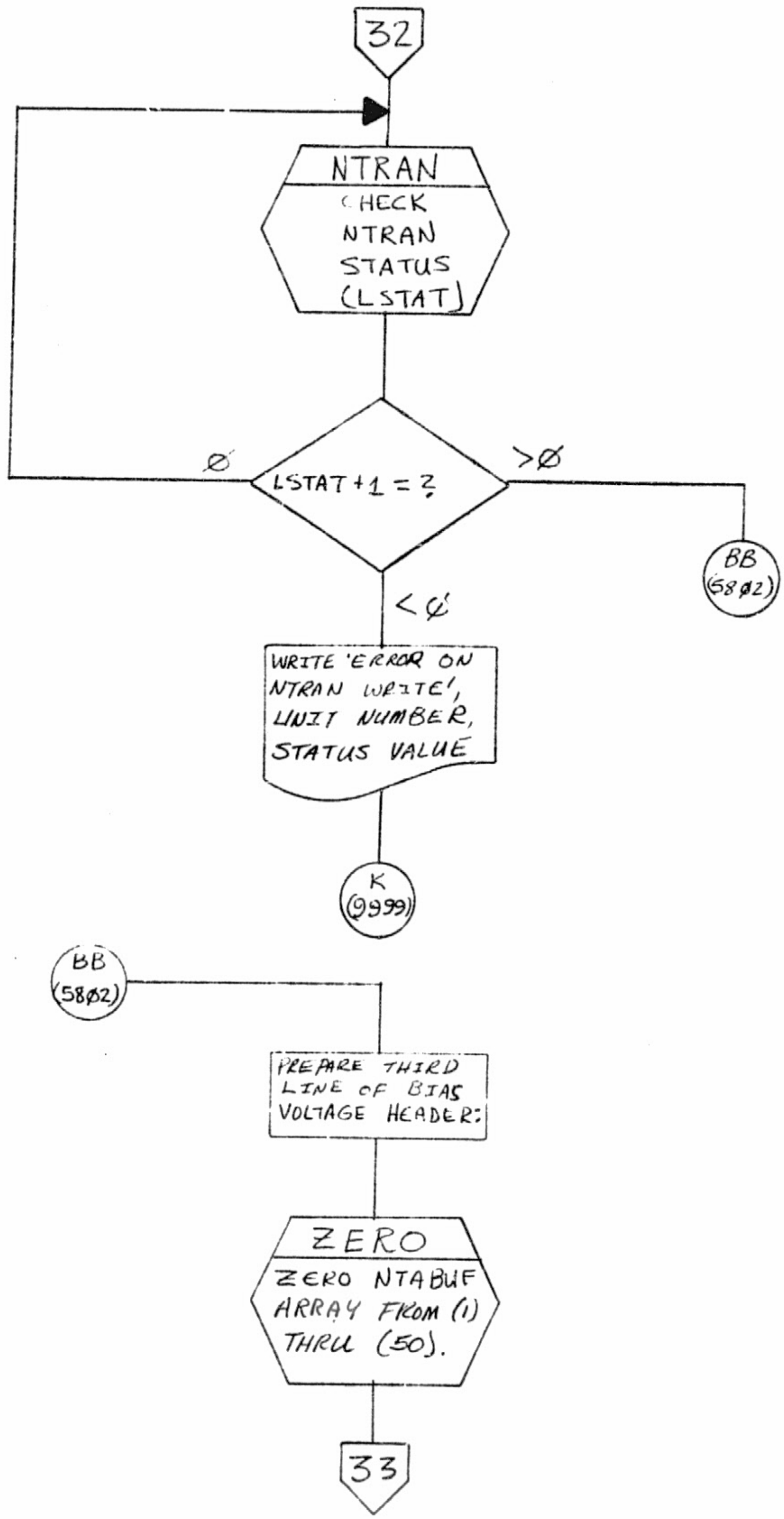

27
Y (52ØØ)
IS DATA OUTSIDE BIAS VOLTAGE PERIOD ?
Y
Z (52ØØ)
N
HAS BIAS VOLTAGE HEADER OR DATA BEEN WRITTEN ON SCRATCH TAPE
Y
AA (525Ø)
N
PREPARE TO WRITE BIAS VOLTAGE HEADER
ZERO
ZERO NTABUF ARRAY FROM (1) THRU (50).
NTABUF(1) = 23
NTABUF(2) = IIN(1Ø)
NTABUF(3) = IIN(9)
NTABUF(4) = IIN(8)
DERIVE BIAS VOLTAGE PERIOD START TIME: TMS153 = SSTIME(1,LBIAS)
28

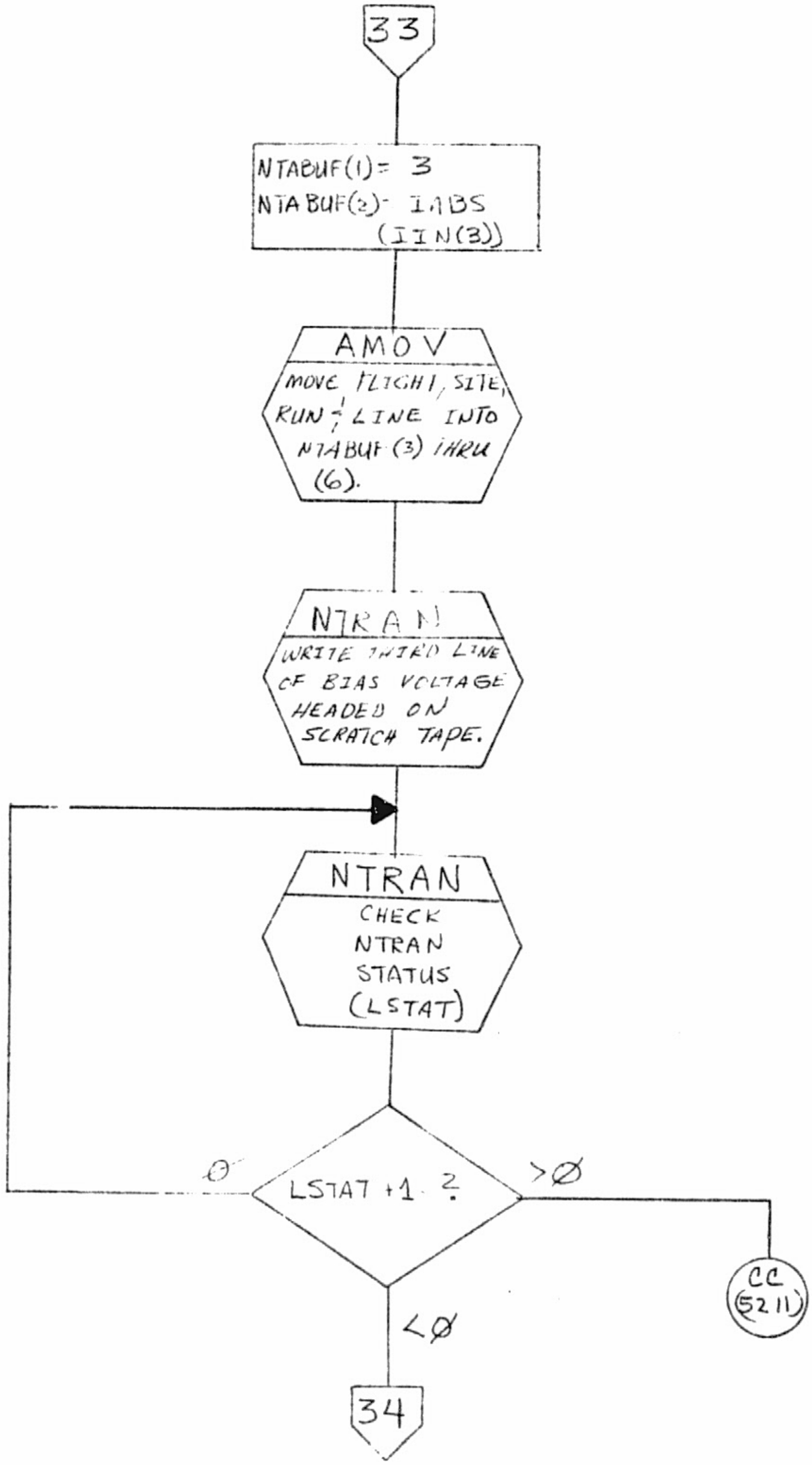

28
STOHMS
CONVERT START TIME TO HOURS MINUTES AND SECONDS
NTABUF(8)= ITIM(1)
NTABUF(9)= ITIM(2)
INTNUM = SEC
RFRAC = SEC - INTNUM
IFRAC = RFRAC * 10000
NTABUF(10) = INTNUM
NTABUF(11) = IFRAC
DERIVE BIAS VOLTAGE PERIOD STOP TIME:
TMS153 = SSTIME (2,LBIAS)
STOHMS
CONVERT STOP TIME TO HOURS, MINUTES, AND SECONDS
NTABUF (12)= ITIM(1)
NTABUF(13)= ITIM(2)
29

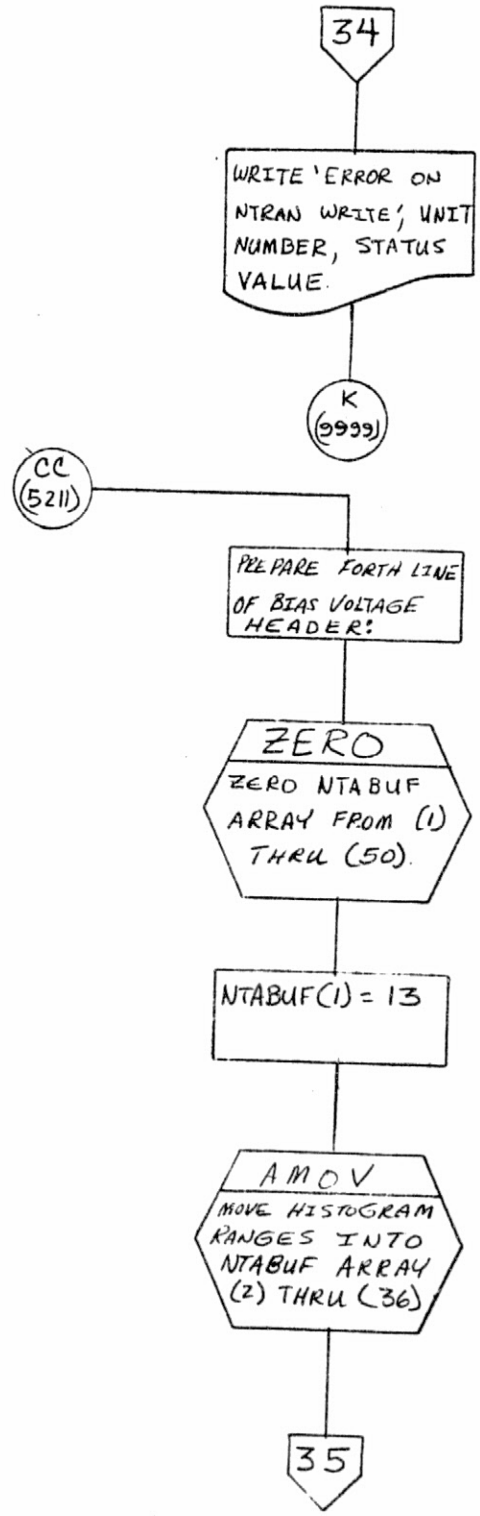

29
INTNUM=SEC
RFRAC=SEC-INTNUM
IFRAC=RFRAC*10000
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
NTRAN
WRITE BIAS VOLTAGE HEADER OUT TO SCRATCH TAPE
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT +1 = ?
Ø
>Ø
<Ø
3Ø
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE.
K
(9999)

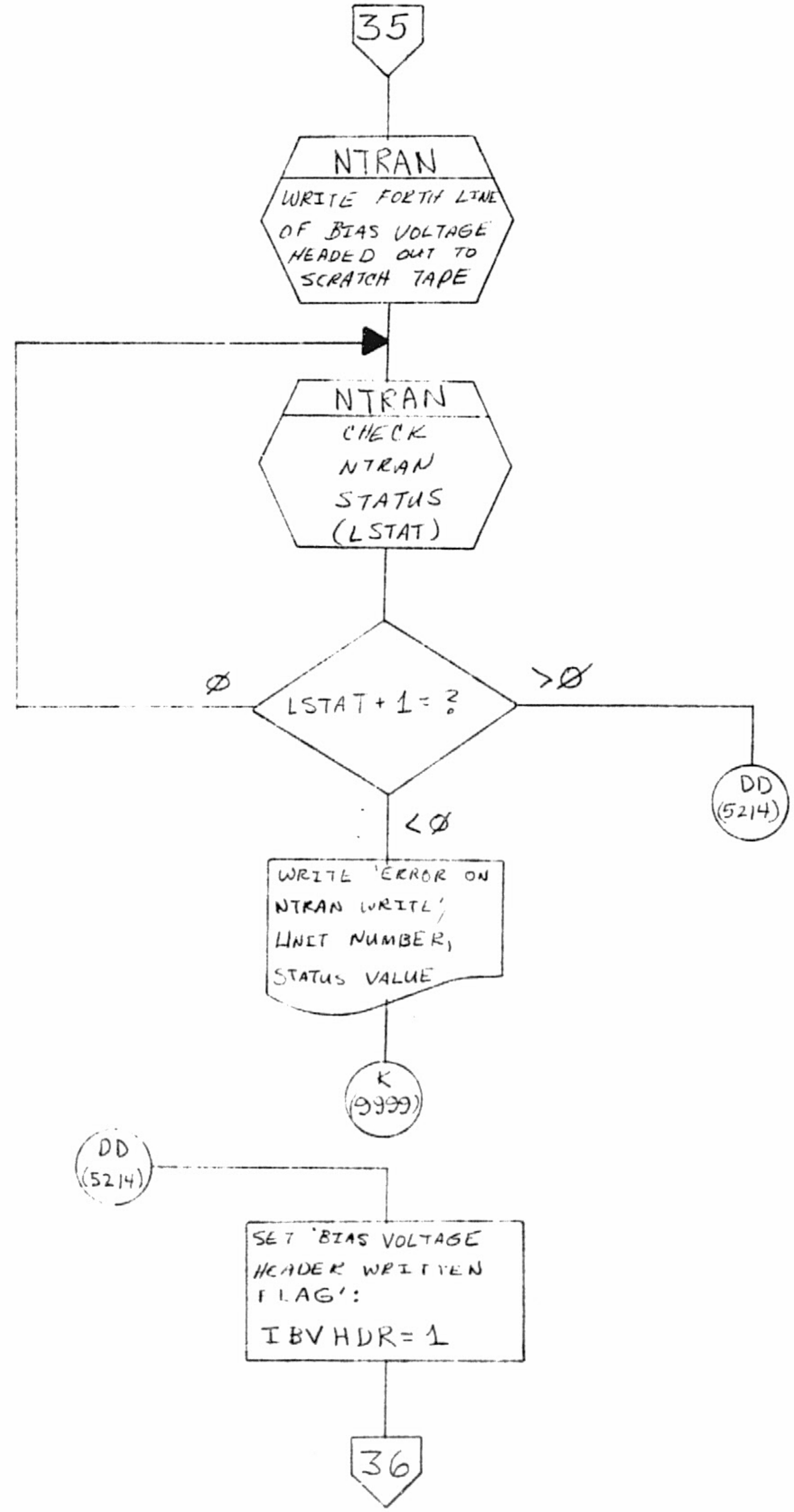

30
PREPARE SECOND LINE OF BIAS VOLTAGE HEADER
ZERO
ZERO NTABUF ARRAY FROM (1) THRU (5Ø)
NTABUF(1) = 2
DERIVE FIRST FRAME AND DELTA TIME:
AMOV
MOVE FIRST FRAME TIME TO D.P. TMS 153
AMOV
MOVE FRAME DELTA TIME TO D.P. DELT.
31

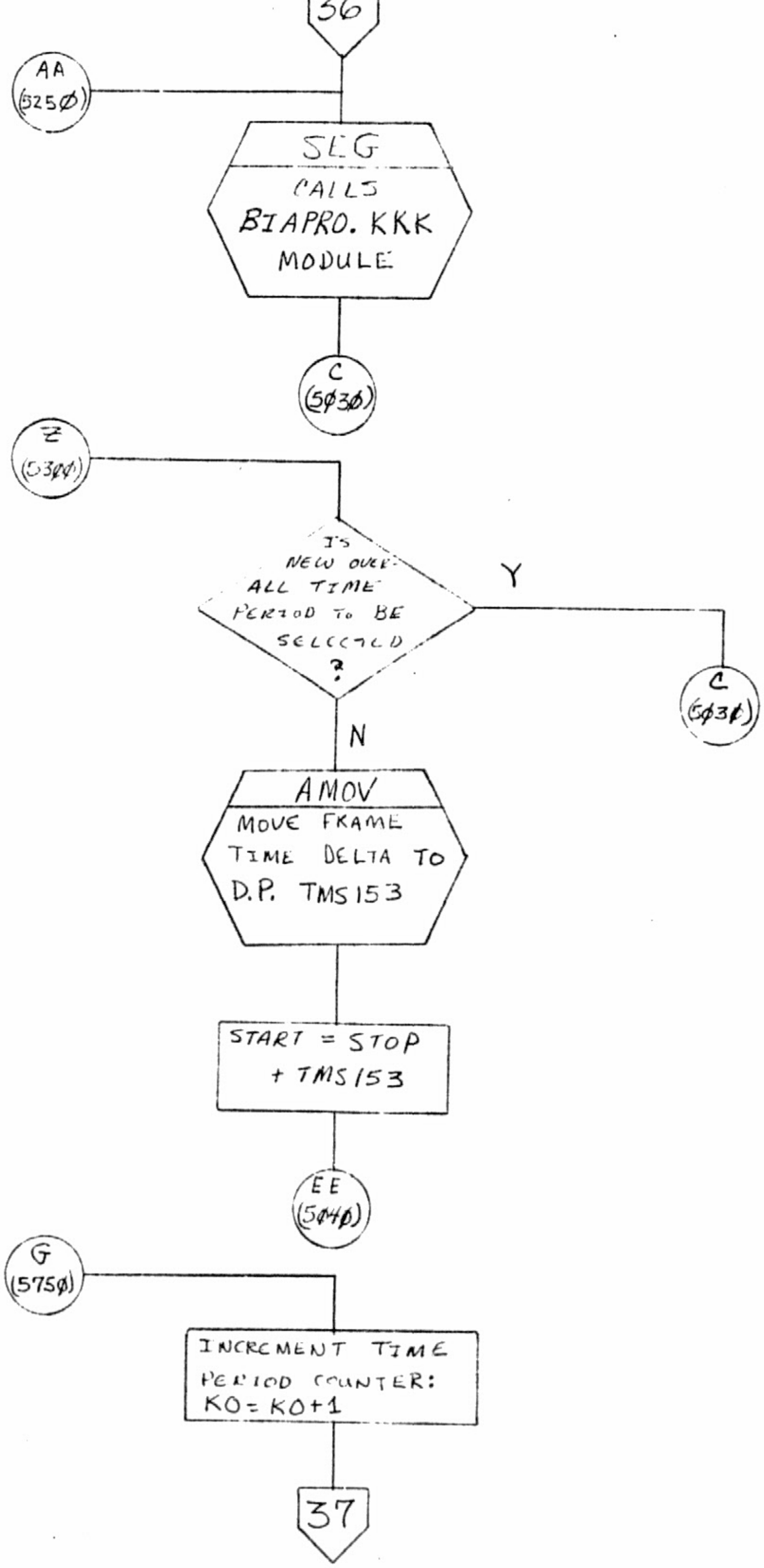

31
IS FIRST FRAME TIME GREATER OR EQUAL TO BIAS VOLTAGE START TIME
Y
N
TMS153 = TMS153 + DELT
STOHMS
CONVERT FRAME TIME TO HOURS, MINUTES, AND SECONDS.
NTABUF(12)=ITIM(1)
NTABUF(13)=ITIM(2)
INTNUM=SEC
RFRAC=SEC-INTNUM
IFRAC=RFRAC*1000
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
NTRAN
WRITE SECOND BIAS VOLTAGE HEADER LINE TO SCRATCH TAPE
32

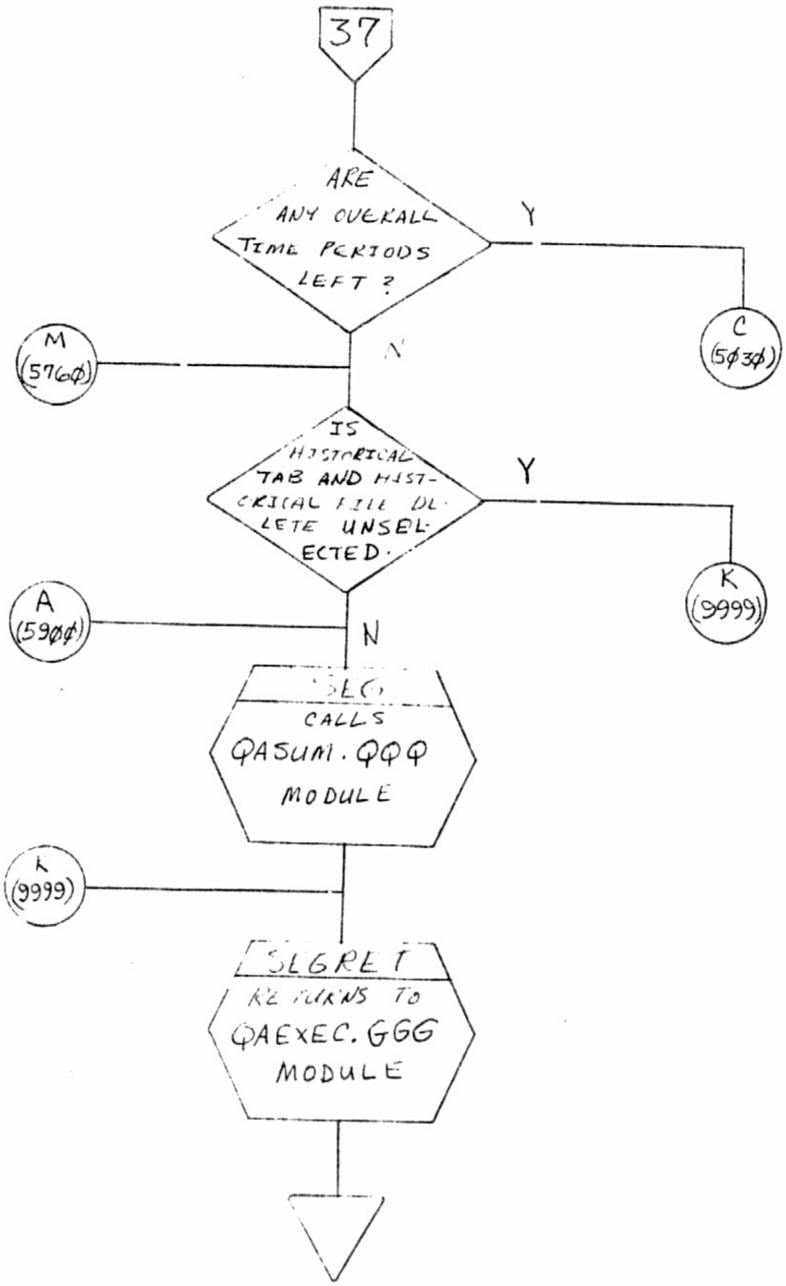

32
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT+1 = ?
Ø
>Ø
<Ø
BB (58Ø2)
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE
K (9999)
BB (58Ø2)
PREPARE THIRD LINE OF BIAS VOLTAGE HEADER:
ZERO
ZERO NTABUF ARRAY FROM (1) THRU (50).
33


33
NTABUF(1) = 3
NTABUF(2) = IABS (IIN(3))
AMOV
MOVE FLIGHT, SITE, RUN ; LINE INTO NTABUF (3) THRU (6).
NTRAN
WRITE THIRD LINE OF BIAS VOLTAGE HEADED ON SCRATCH TAPE.
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT +1 ?
0
>Ø
<Ø
CC (52 11)
34

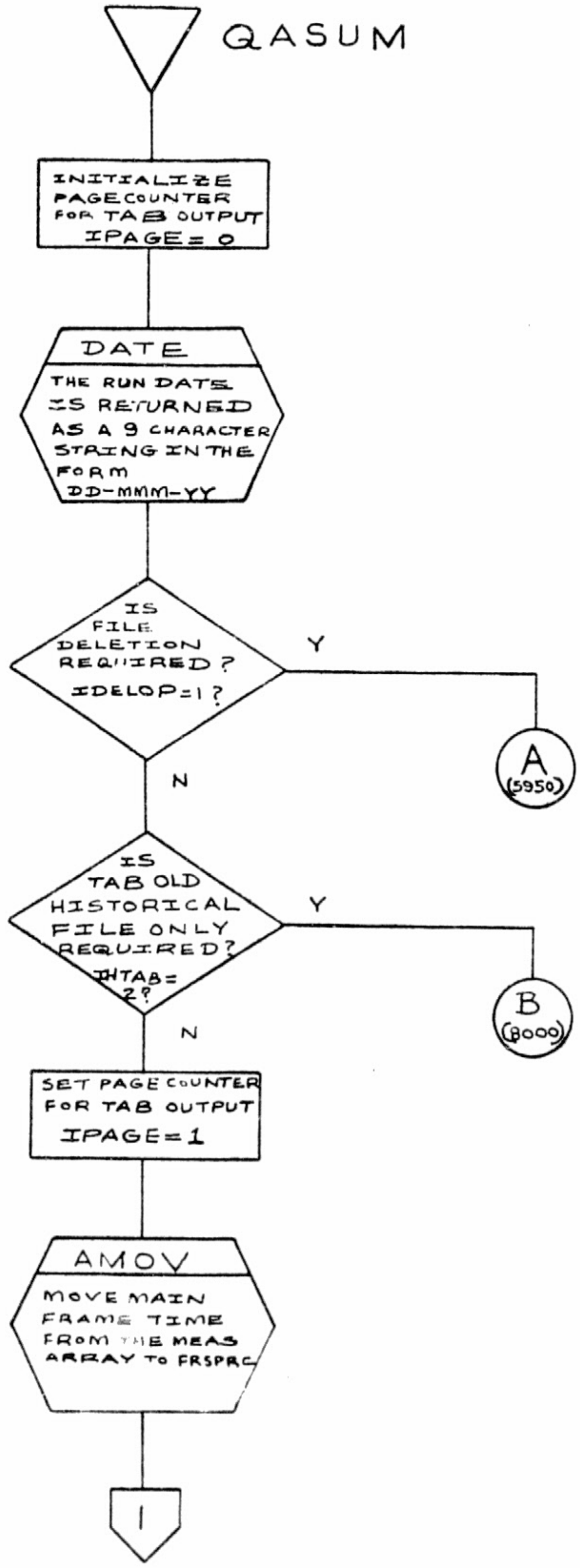

34
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE.
K (9999)
CC (5211)
PREPARE FORTH LINE OF BIAS VOLTAGE HEADER.
ZERO
ZERO NTABUF ARRAY FROM (1) THRU (50).
NTABUF(1) = 13
AMOV
MOVE HISTOGRAM RANGES INTO NTABUF ARRAY (2) THRU (36)
35

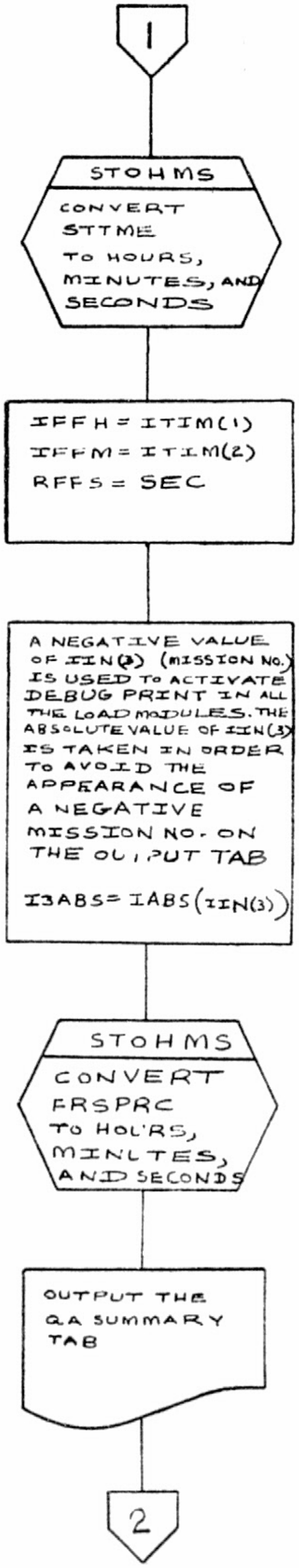

35
NTRAN
WRITE FORTH LINE OF BIAS VOLTAGE HEADED OUT TO SCRATCH TAPE
NTRAN
CHECK NTRAN STATUS (LSTAT)
Ø
LSTAT + 1 = ?
>Ø
<Ø
DD (5214)
WRITE 'ERROR ON NTRAN WRITE', UNIT NUMBER, STATUS VALUE
K (9999)
DD (5214)
SET 'BIAS VOLTAGE HEADER WRITTEN FLAG':
IBVHDR = 1
36

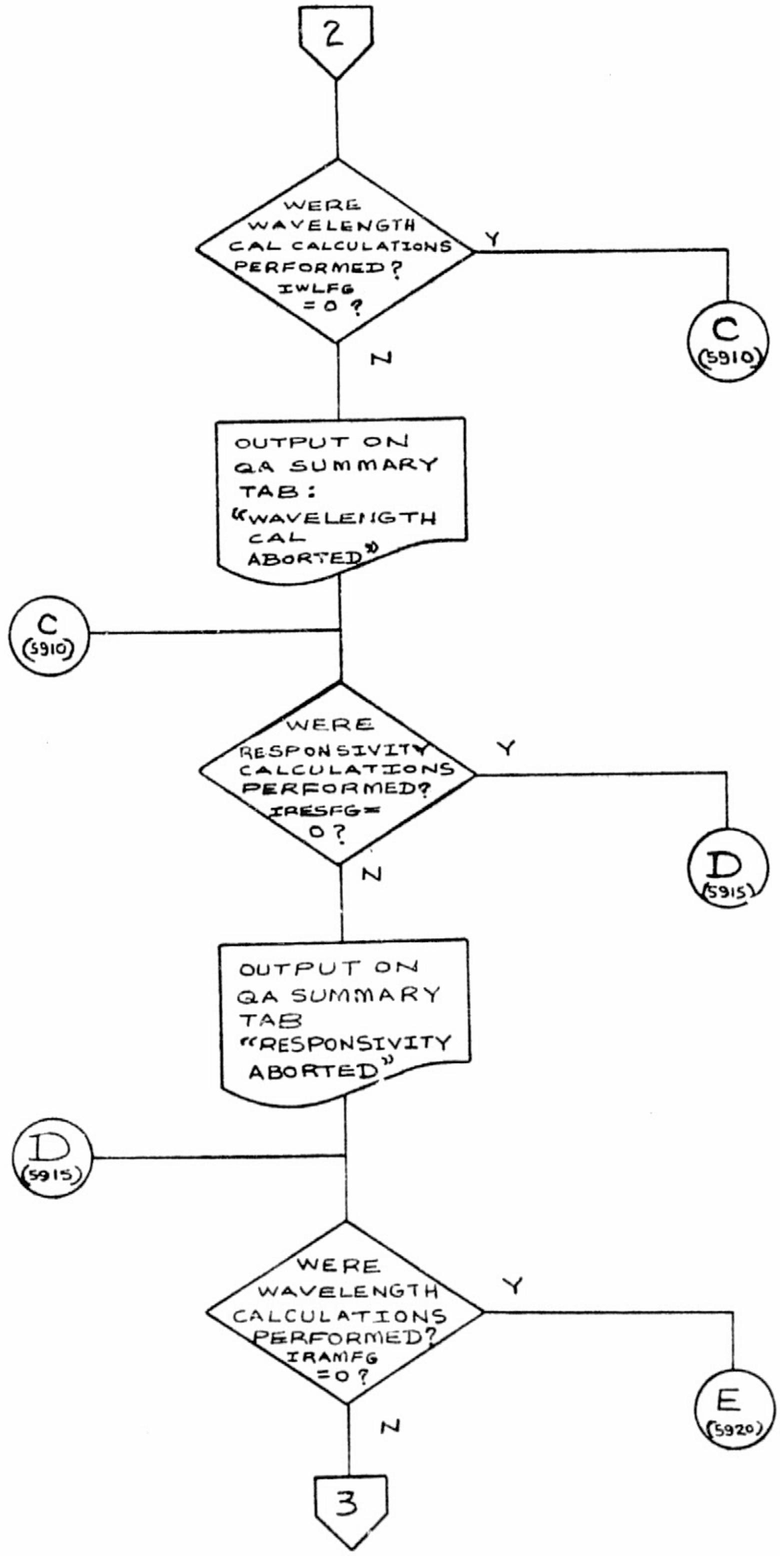

36
AA (525Ø)
SEG
CALLS BIAPRO. KKK MODULE
C (5Ø3Ø)
Z (53ØØ)
IS NEW OVER ALL TIME PERIOD TO BE SELECTED ?
Y
C (5Ø3Ø)
N
AMOV
MOVE FRAME TIME DELTA TO D.P. TMS153
START = STOP + TMS153
EE (5Ø4Ø)
G (575Ø)
INCREMENT TIME PERIOD COUNTER: KO=KO+1
37

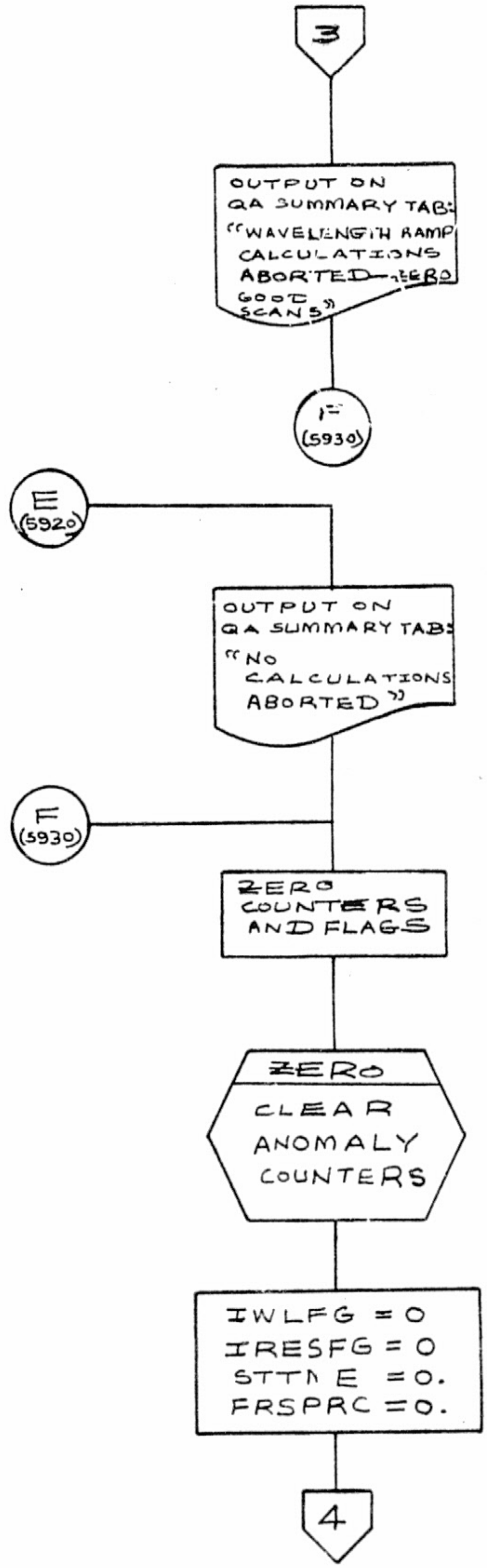


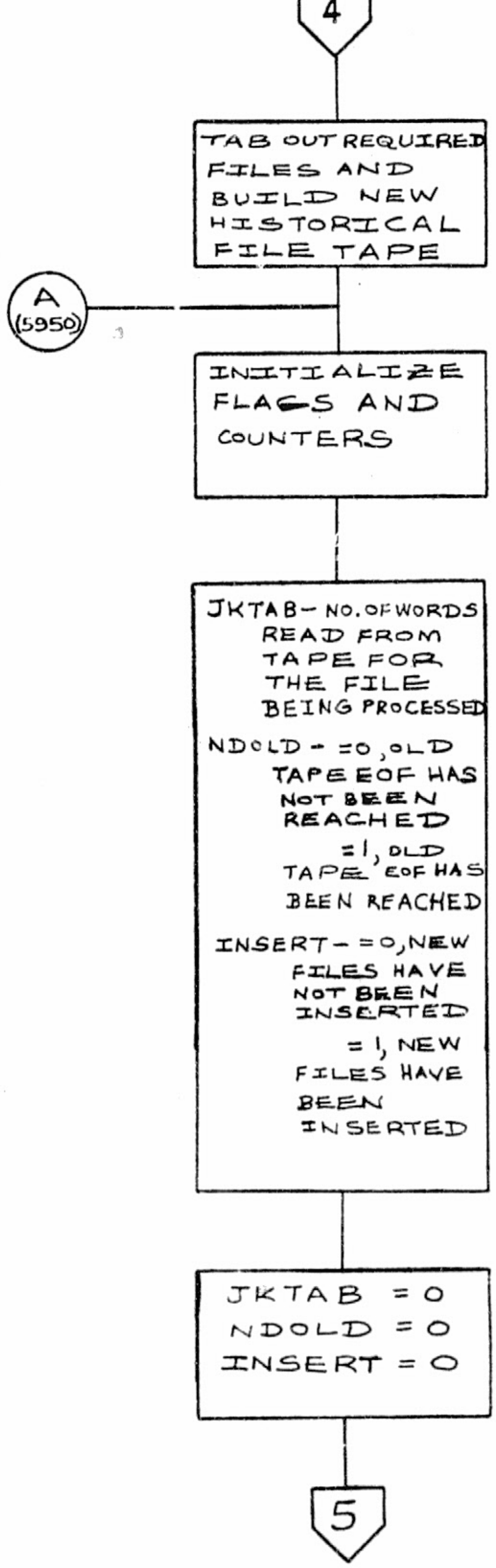
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

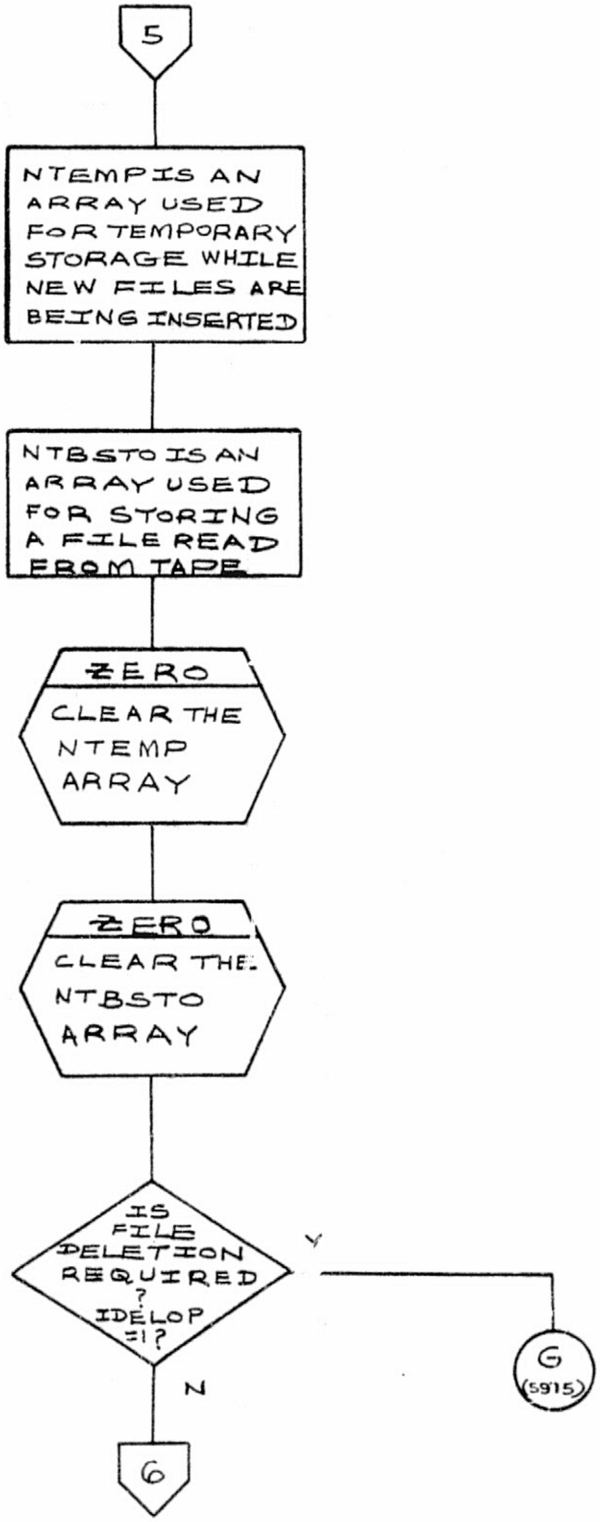

QASUM
INITIALIZE PAGECOUNTER FOR TAB OUTPUT IPAGE=0
DATE
THE RUN DATE IS RETURNED AS A 9 CHARACTER STRING IN THE FORM DD-MMM-YY
IS FILE DELETION REQUIRED? IDELOP=1?
Y
A (5950)
N
IS TAB OLD HISTORICAL FILE ONLY REQUIRED? IHTAB=2?
Y
B (8000)
N
SET PAGE COUNTER FOR TAB OUTPUT IPAGE=1
AMOV
MOVE MAIN FRAME TIME FROM THE MEAS ARRAY TO FRSPRC
1

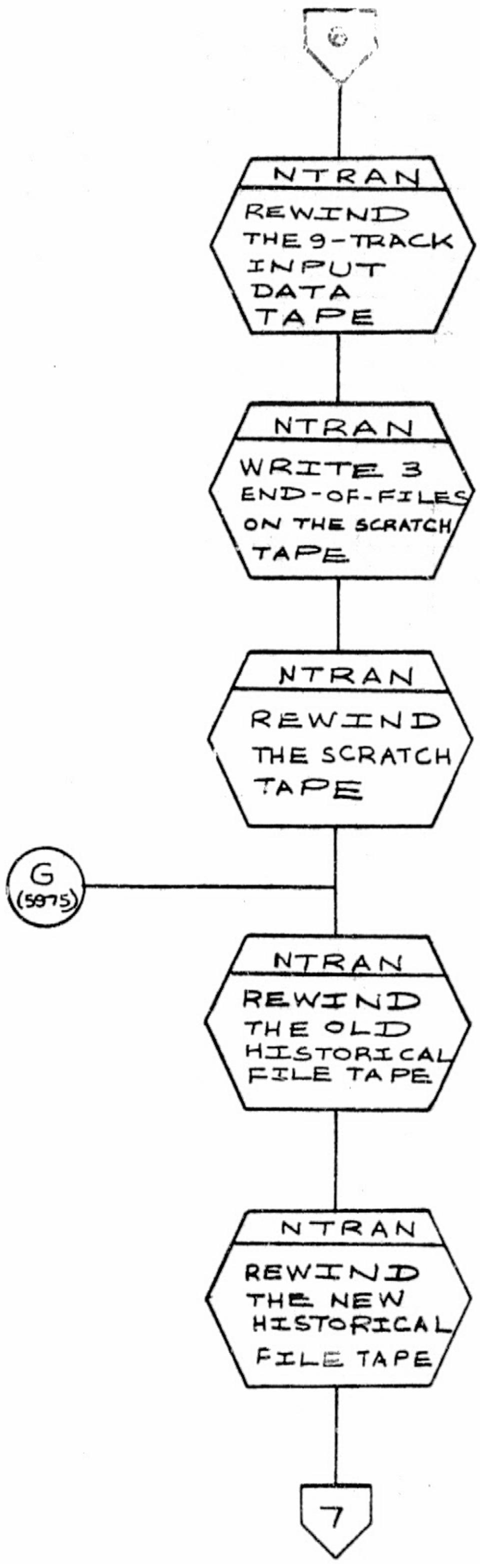

1
STOHMS
CONVERT STTME TO HOURS, MINUTES, AND SECONDS
IFFH = ITIM(1)
IFFM = ITIM(2)
RFFS = SEC
A NEGATIVE VALUE OF IIN(3) (MISSION NO.) IS USED TO ACTIVATE DEBUG PRINT IN ALL THE LOAD MODULES. THE ABSOLUTE VALUE OF IIN(3) IS TAKEN IN ORDER TO AVOID THE APPEARANCE OF A NEGATIVE MISSION NO. ON THE OUTPUT TAB
I3ABS= IABS(IIN(3))
STOHMS
CONVERT FRSPRC TO HOURS, MINUTES, AND SECONDS
OUTPUT THE QA SUMMARY TAB
2

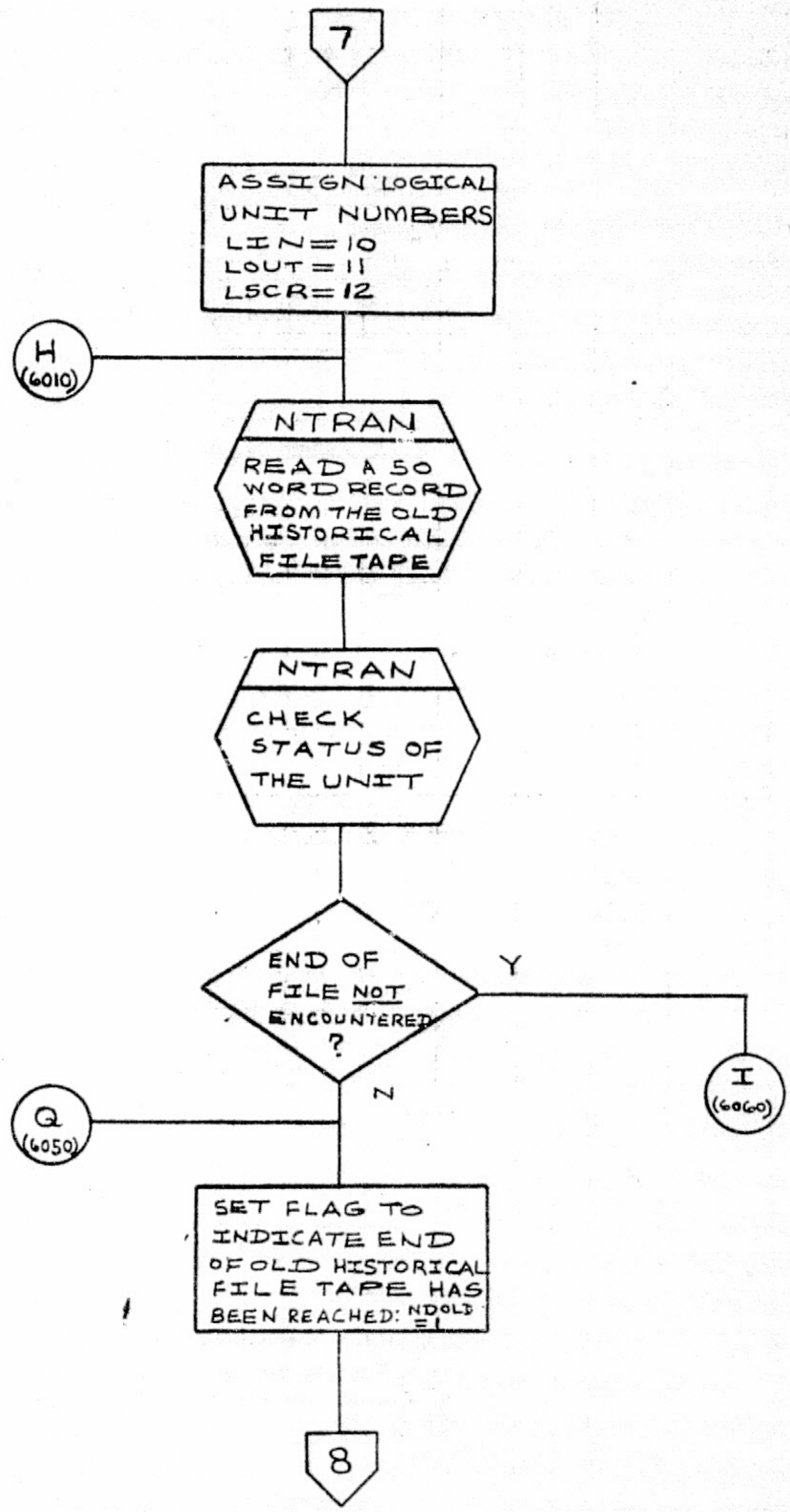

2
WERE WAVELENGTH CAL CALCULATIONS PERFORMED? IWLFG =0 ?
Y
C (5910)
N
OUTPUT ON QA SUMMARY TAB: "WAVELENGTH CAL ABORTED"
C (5910)
WERE RESPONSIVITY CALCULATIONS PERFORMED? IRESFG= 0 ?
Y
D (5915)
N
OUTPUT ON QA SUMMARY TAB "RESPONSIVITY ABORTED"
D (5915)
WERE WAVELENGTH CALCULATIONS PERFORMED? IRAMFG =0 ?
Y
E (5920)
N
3

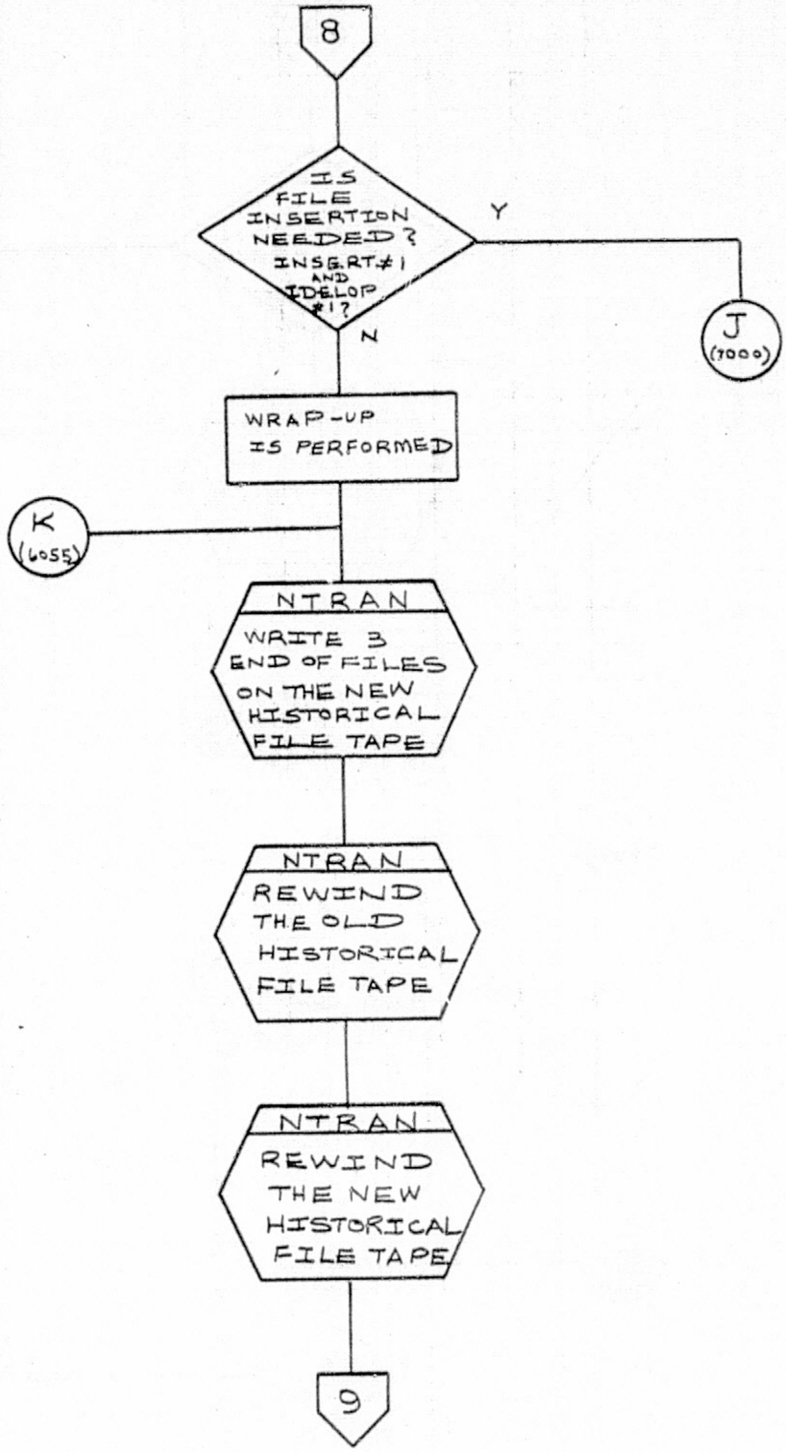

3
OUTPUT ON QA SUMMARY TAB: "WAVELENGTH RAMP CALCULATIONS ABORTED—ZERO GOOD SCANS"
F (5930)
E (5920)
OUTPUT ON QA SUMMARY TAB: "NO CALCULATIONS ABORTED"
F (5930)
ZERO COUNTERS AND FLAGS
ZERO
CLEAR ANOMALY COUNTERS
IWLFG = 0
IRESFG = 0
STTME = 0.
FRSPRC = 0.
4

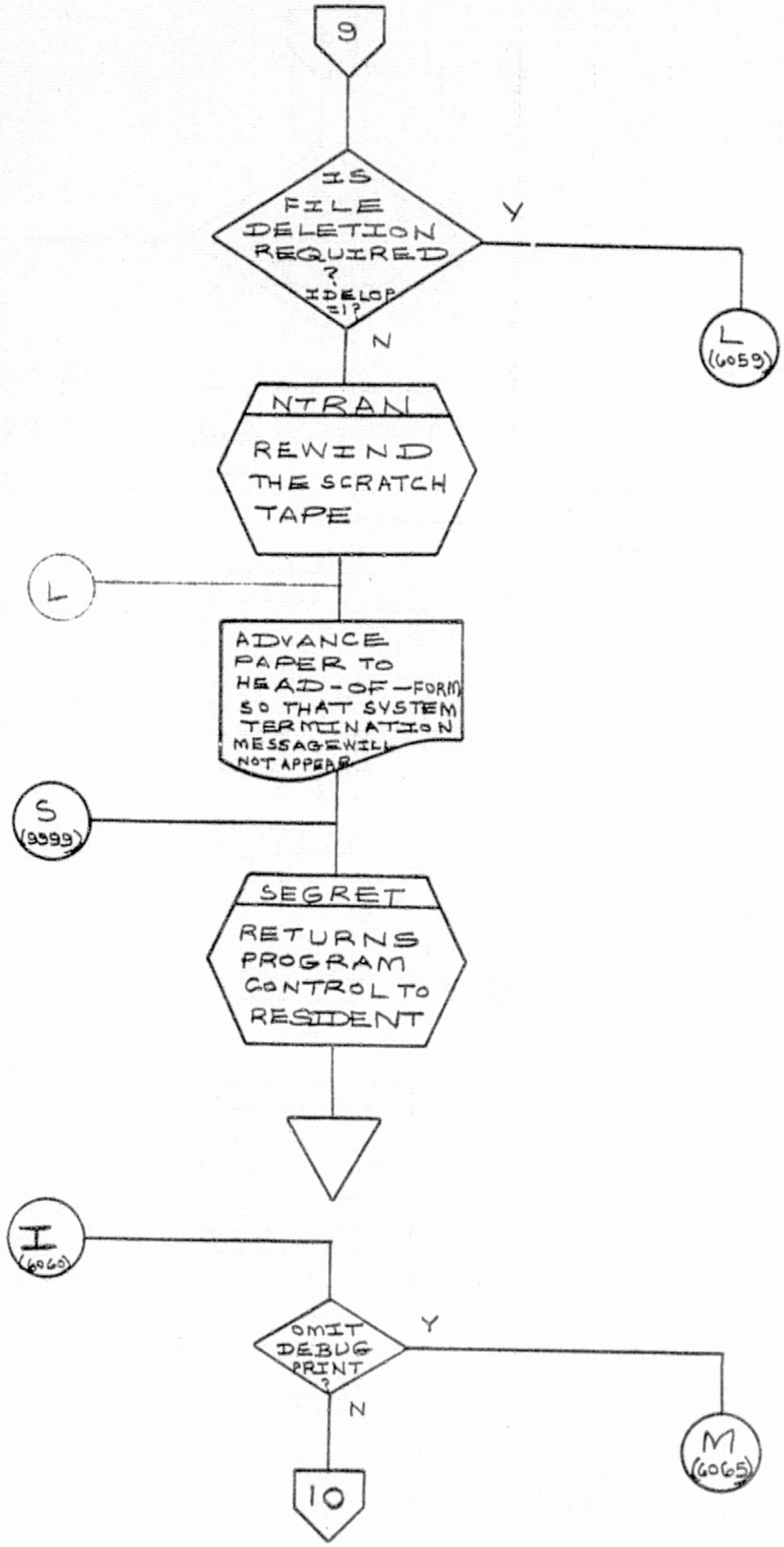

4
TAB OUT REQUIRED FILES AND BUILD NEW HISTORICAL FILE TAPE
A (5950)
INITIALIZE FLAGS AND COUNTERS
JKTAB – NO. OF WORDS READ FROM TAPE FOR THE FILE BEING PROCESSED
NDOLD – =0, OLD TAPE EOF HAS NOT BEEN REACHED
=1, OLD TAPE EOF HAS BEEN REACHED
INSERT – =0, NEW FILES HAVE NOT BEEN INSERTED
=1, NEW FILES HAVE BEEN INSERTED
JKTAB = 0
NDOLD = 0
INSERT = 0
5

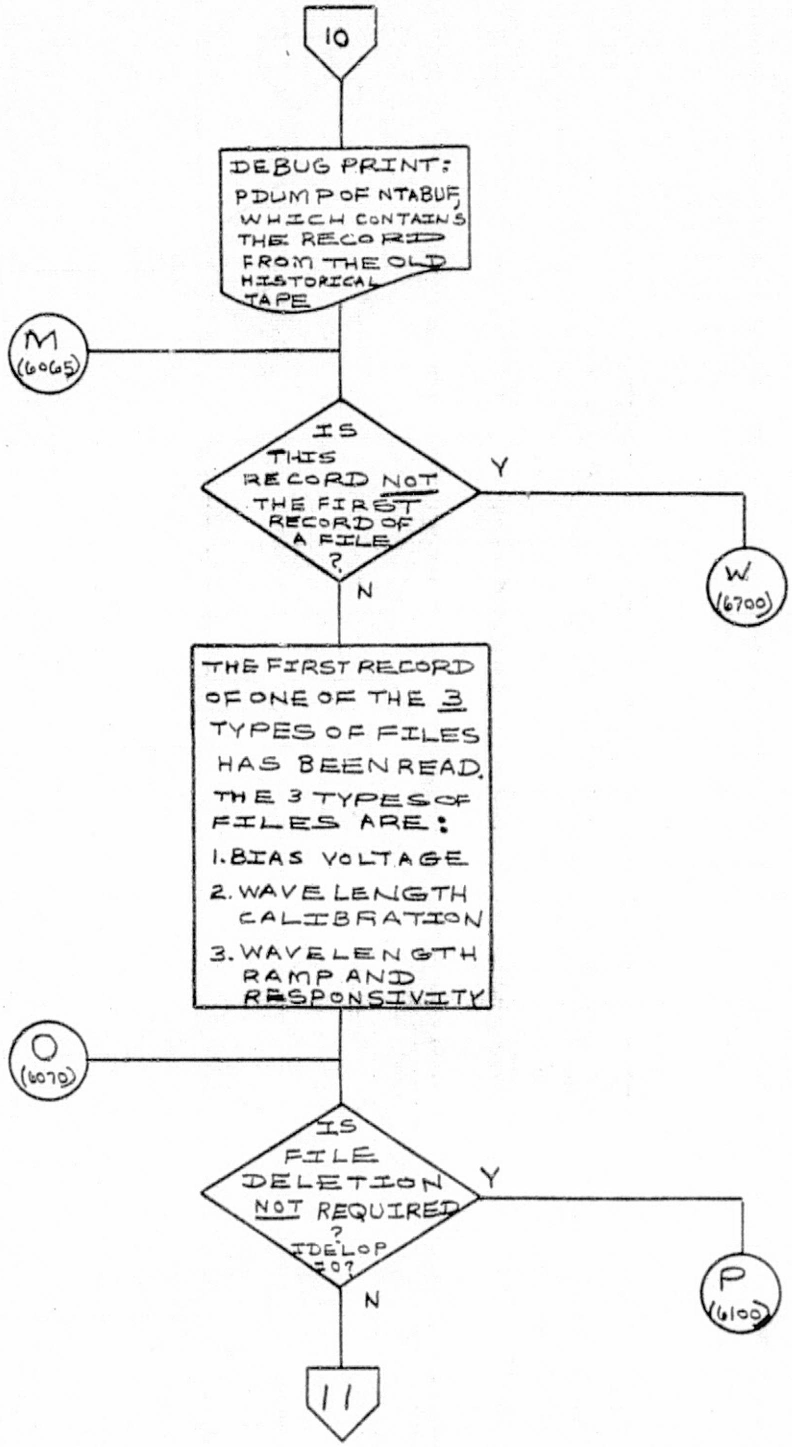

5
NTEMP IS AN ARRAY USED FOR TEMPORARY STORAGE WHILE NEW FILES ARE BEING INSERTED
NTBSTO IS AN ARRAY USED FOR STORING A FILE READ FROM TAPE
ZERO
CLEAR THE NTEMP ARRAY
ZERO
CLEAR THE NTBSTO ARRAY
IS FILE DELETION REQUIRED ? IDELOP =1?
Y
N
G (5915)
6

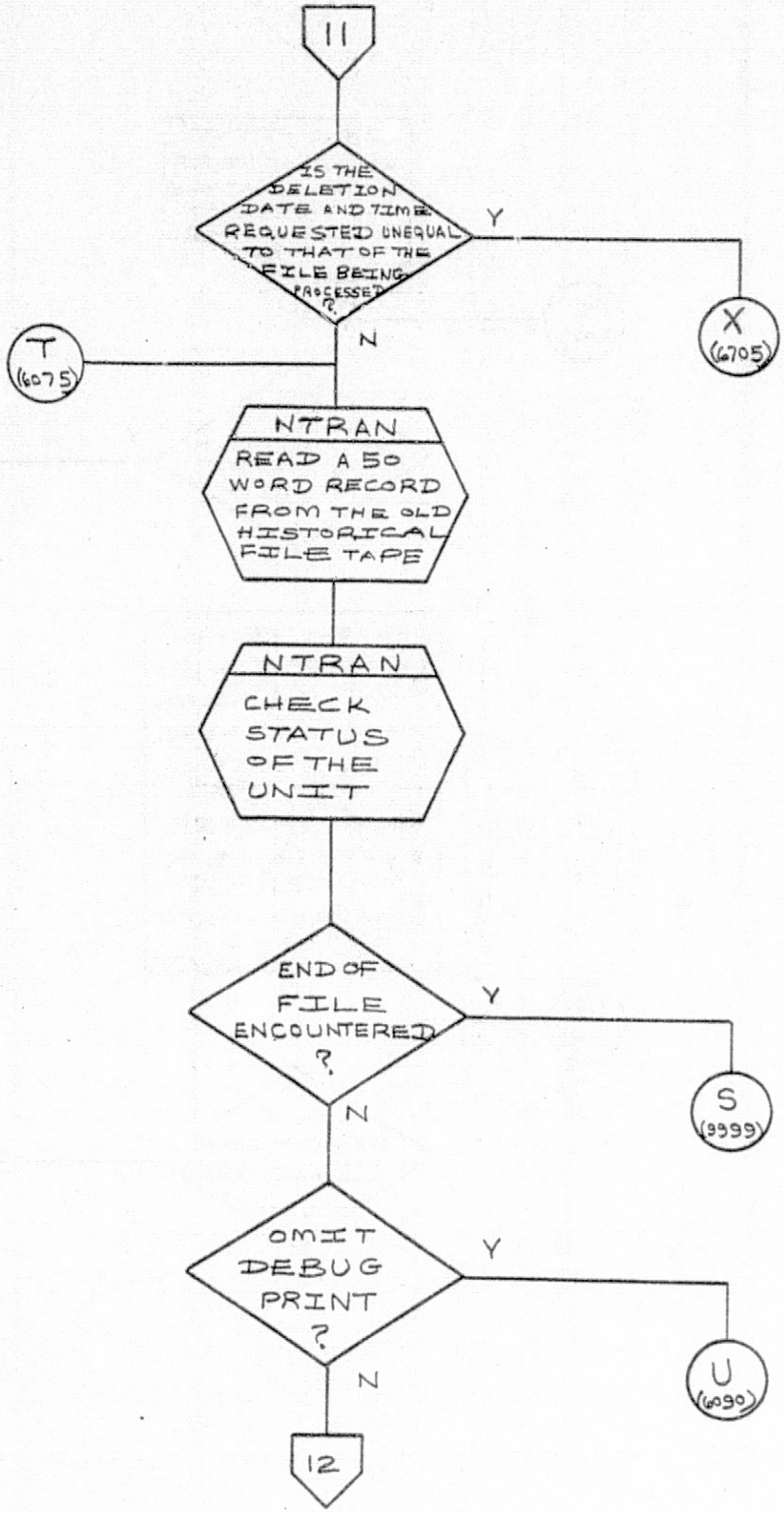

6
NTRAN
REWIND THE 9-TRACK INPUT DATA TAPE
NTRAN
WRITE 3 END-OF-FILES ON THE SCRATCH TAPE
NTRAN
REWIND THE SCRATCH TAPE
G (5975)
NTRAN
REWIND THE OLD HISTORICAL FILE TAPE
NTRAN
REWIND THE NEW HISTORICAL FILE TAPE
7

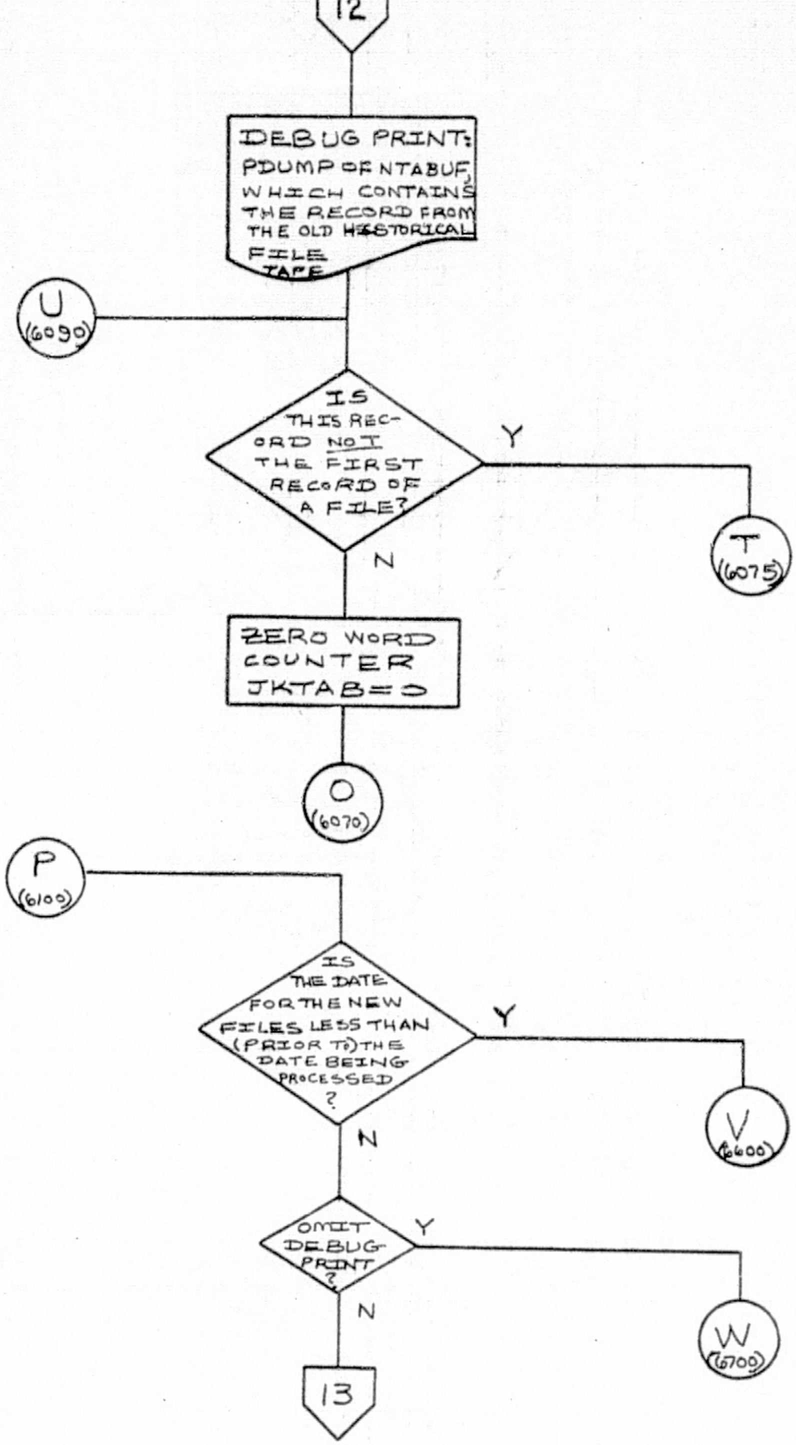

7
ASSIGN LOGICAL UNIT NUMBERS
LIN=10
LOUT=11
LSCR=12
H (6010)
NTRAN
READ A 50 WORD RECORD FROM THE OLD HISTORICAL FILE TAPE
NTRAN
CHECK STATUS OF THE UNIT
END OF FILE NOT ENCOUNTERED ?
Y
N
I (6060)
Q (6050)
SET FLAG TO INDICATE END OF OLD HISTORICAL FILE TAPE HAS BEEN REACHED: NDOLD =1
8

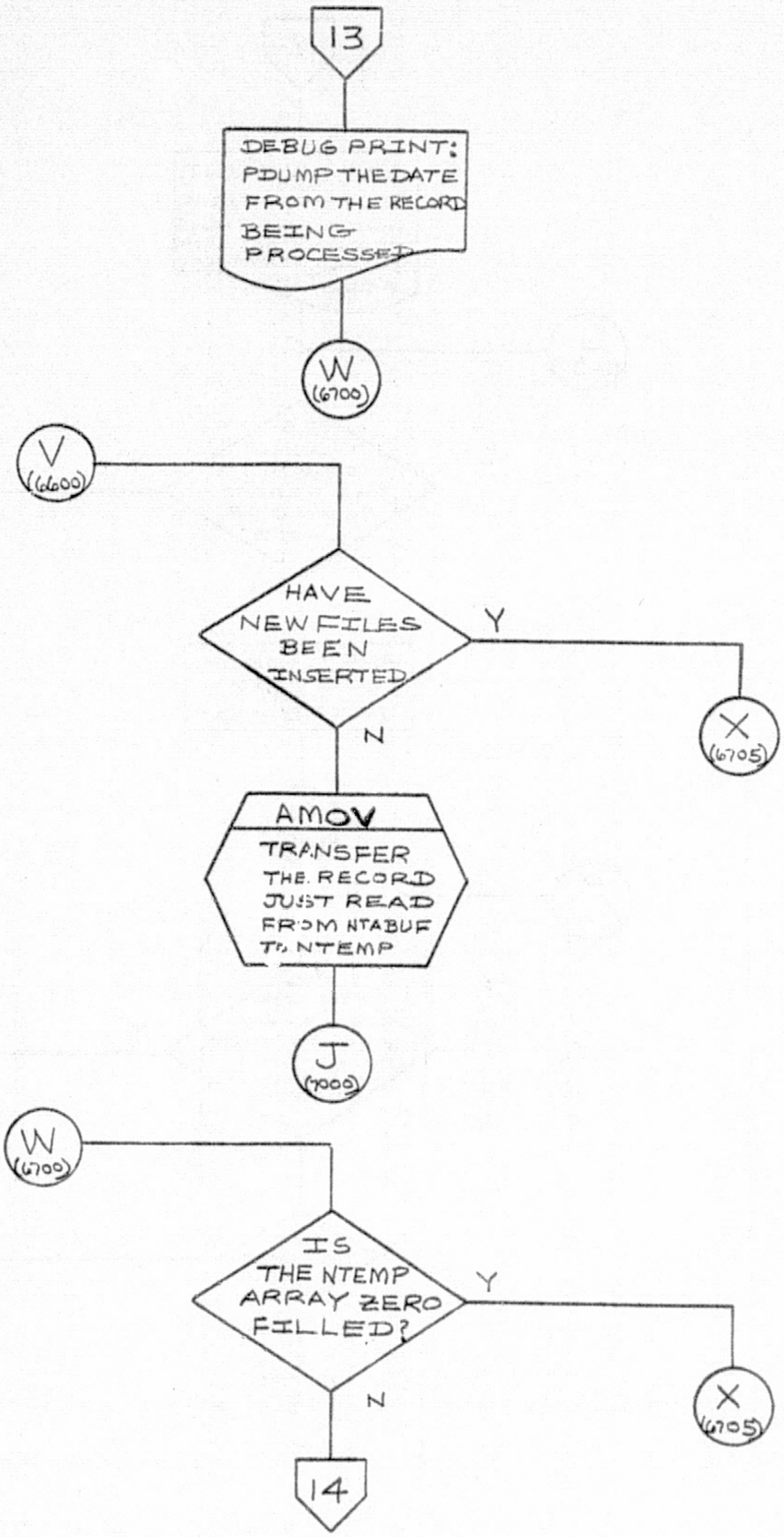

8
IS FILE INSERTION NEEDED? INSERT ≠ 1 AND IDELOP ≠ 1?
Y
N
J (7000)
WRAP-UP IS PERFORMED
K (6055)
NTRAN
WRITE 3 END OF FILES ON THE NEW HISTORICAL FILE TAPE
NTRAN
REWIND THE OLD HISTORICAL FILE TAPE
NTRAN
REWIND THE NEW HISTORICAL FILE TAPE
9

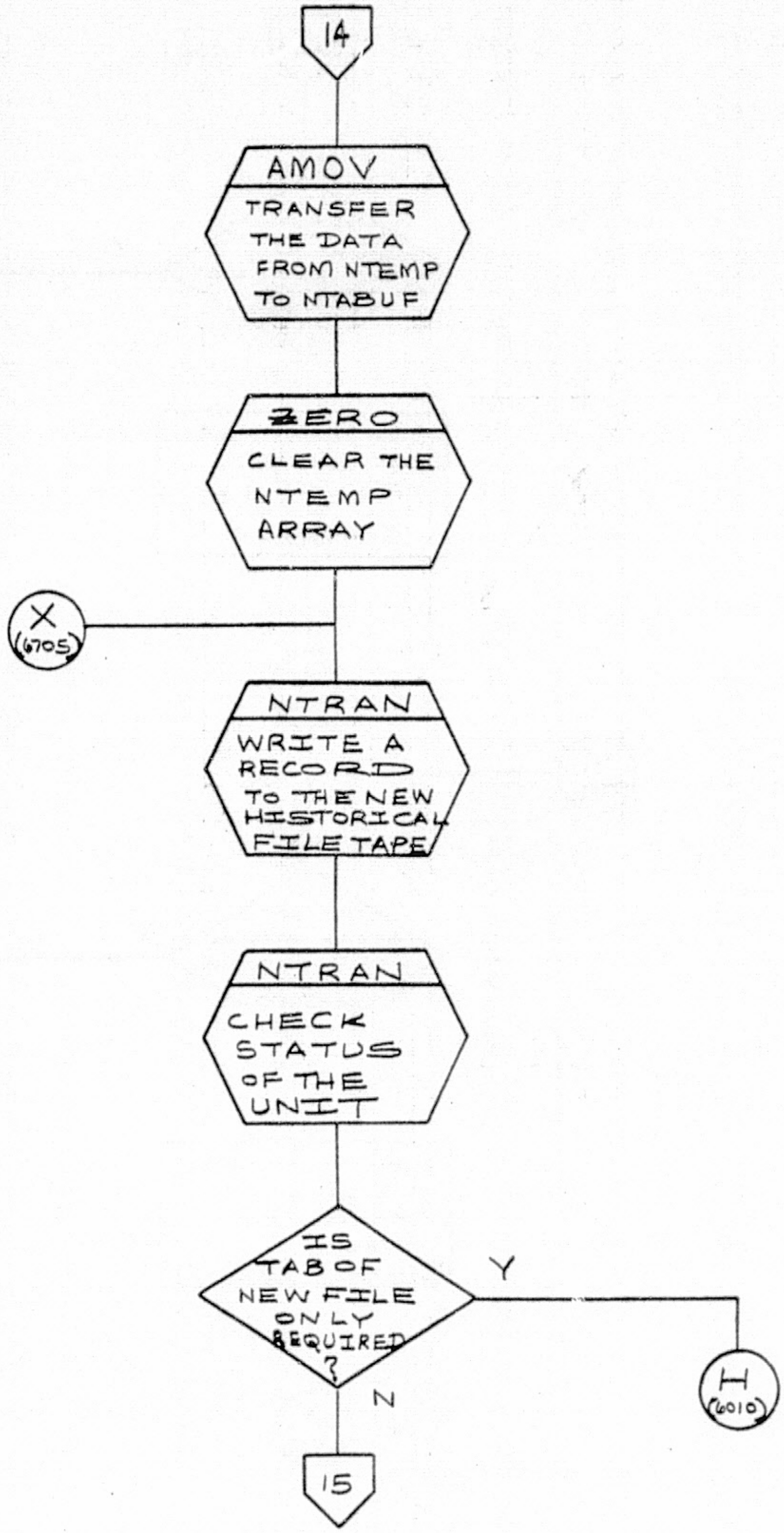

9
IS FILE DELETION REQUIRED ? IDELOP =1?
Y
N
L (6059)
NTRAN
REWIND THE SCRATCH TAPE
L
ADVANCE PAPER TO HEAD-OF-FORM SO THAT SYSTEM TERMINATION MESSAGE WILL NOT APPEAR
S (9999)
SEGRET
RETURNS PROGRAM CONTROL TO RESIDENT
I (6060)
OMIT DEBUG PRINT ?
Y
N
M (6065)
10

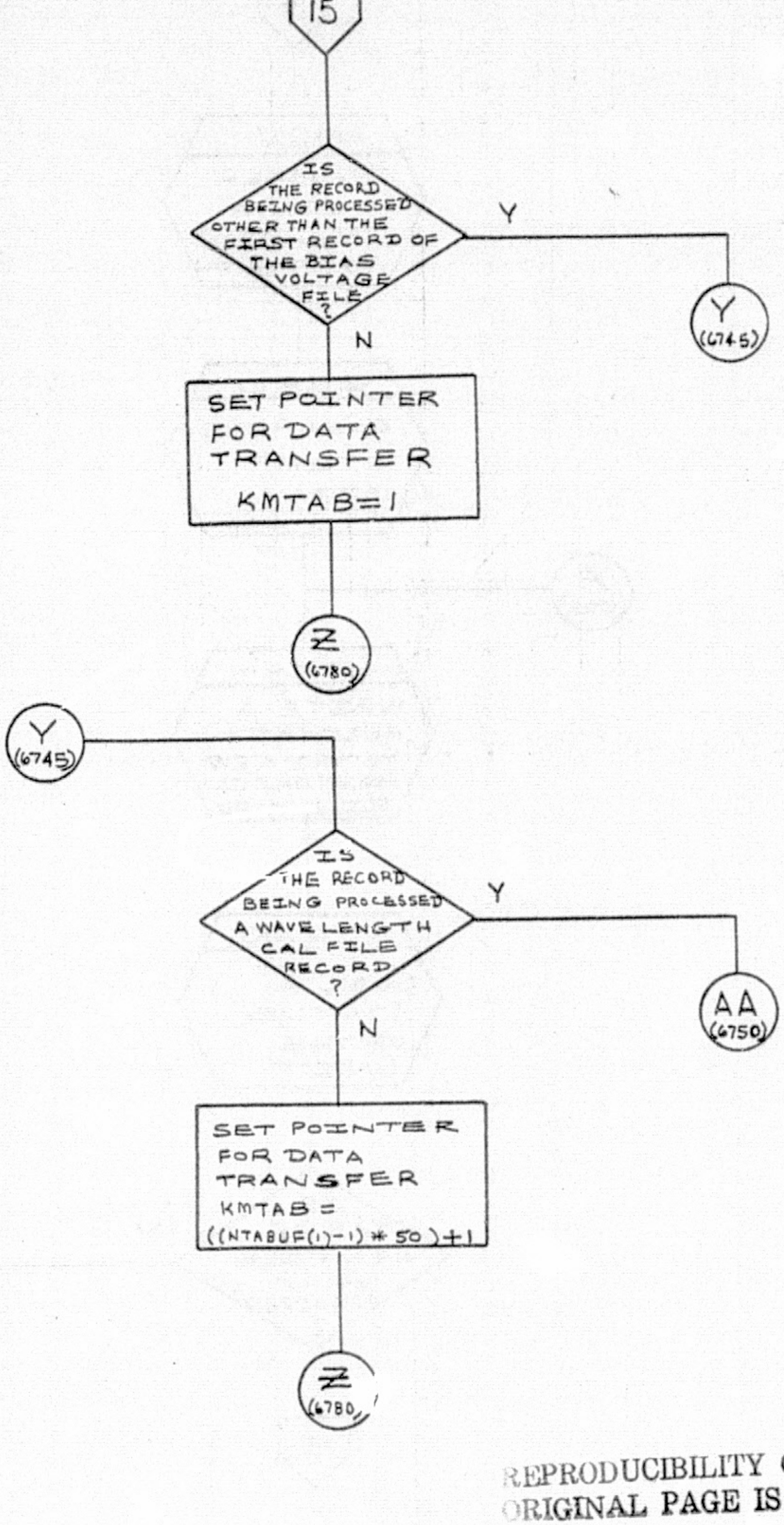

10
DEBUG PRINT: PDUMP OF NTABUF, WHICH CONTAINS THE RECORD FROM THE OLD HISTORICAL TAPE
M (6065)
IS THIS RECORD NOT THE FIRST RECORD OF A FILE ?
Y
W (6700)
N
THE FIRST RECORD OF ONE OF THE 3 TYPES OF FILES HAS BEEN READ. THE 3 TYPES OF FILES ARE:
1. BIAS VOLTAGE
2. WAVE LENGTH CALIBRATION
3. WAVELENGTH RAMP AND RESPONSIVITY
O (6070)
IS FILE DELETION NOT REQUIRED ? IDELOP =0?
Y
P (6100)
N
11

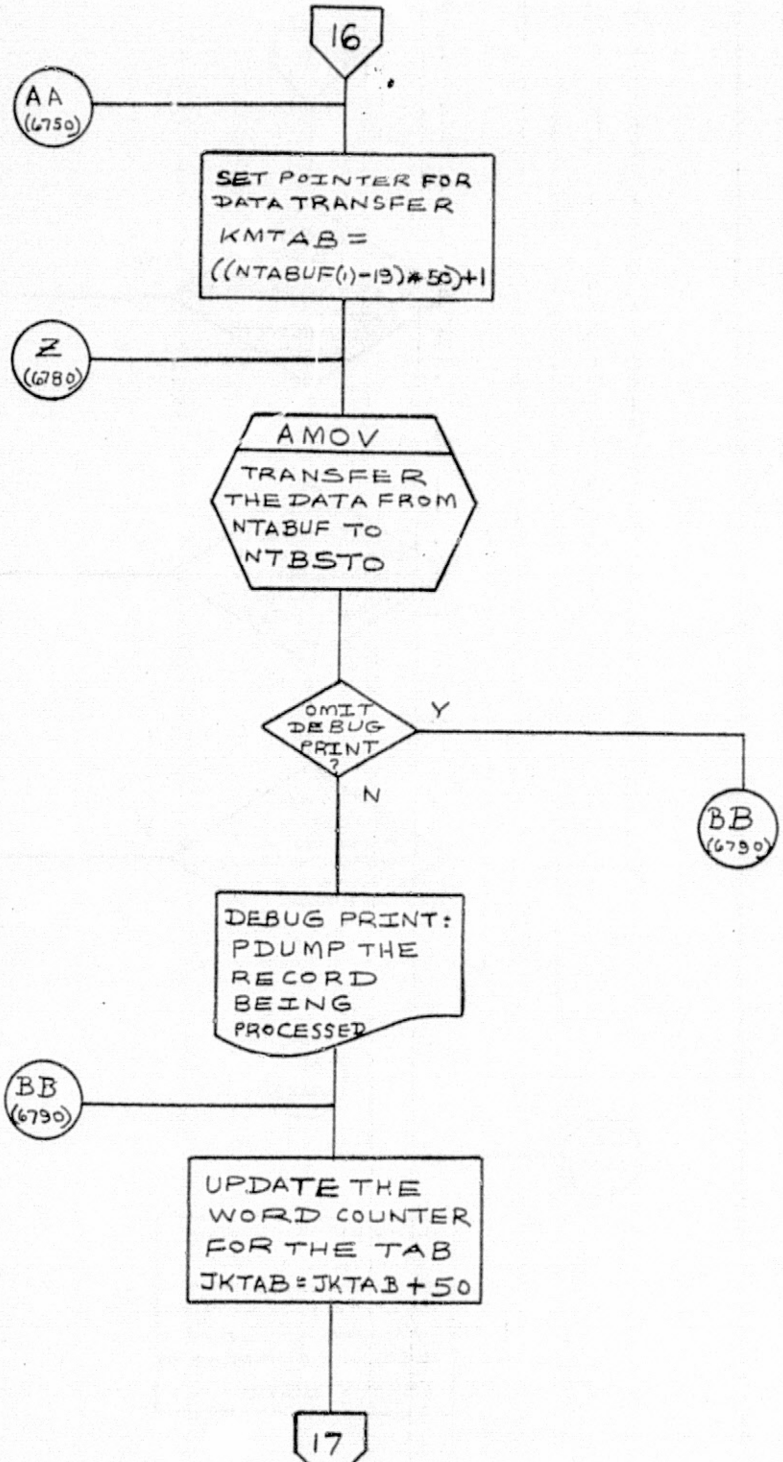

11
IS THE DELETION DATE AND TIME REQUESTED UNEQUAL TO THAT OF THE FILE BEING PROCESSED ?
Y
X (6705)
N
T (6075)
NTRAN
READ A 50 WORD RECORD FROM THE OLD HISTORICAL FILE TAPE
NTRAN
CHECK STATUS OF THE UNIT
END OF FILE ENCOUNTERED ?
Y
S (9999)
N
OMIT DEBUG PRINT ?
Y
U (6090)
N
12


12
DEBUG PRINT: PDUMP OF NTABUF, WHICH CONTAINS THE RECORD FROM THE OLD HISTORICAL FILE TAPE
U (6090)
IS THIS RECORD NOT THE FIRST RECORD OF A FILE?
Y
T (6075)
N
ZERO WORD COUNTER JKTAB=0
O (6070)
P (6100)
IS THE DATE FOR THE NEW FILES LESS THAN (PRIOR TO) THE DATE BEING PROCESSED ?
Y
V (6600)
N
OMIT DEBUG PRINT ?
Y
W (6700)
N
13

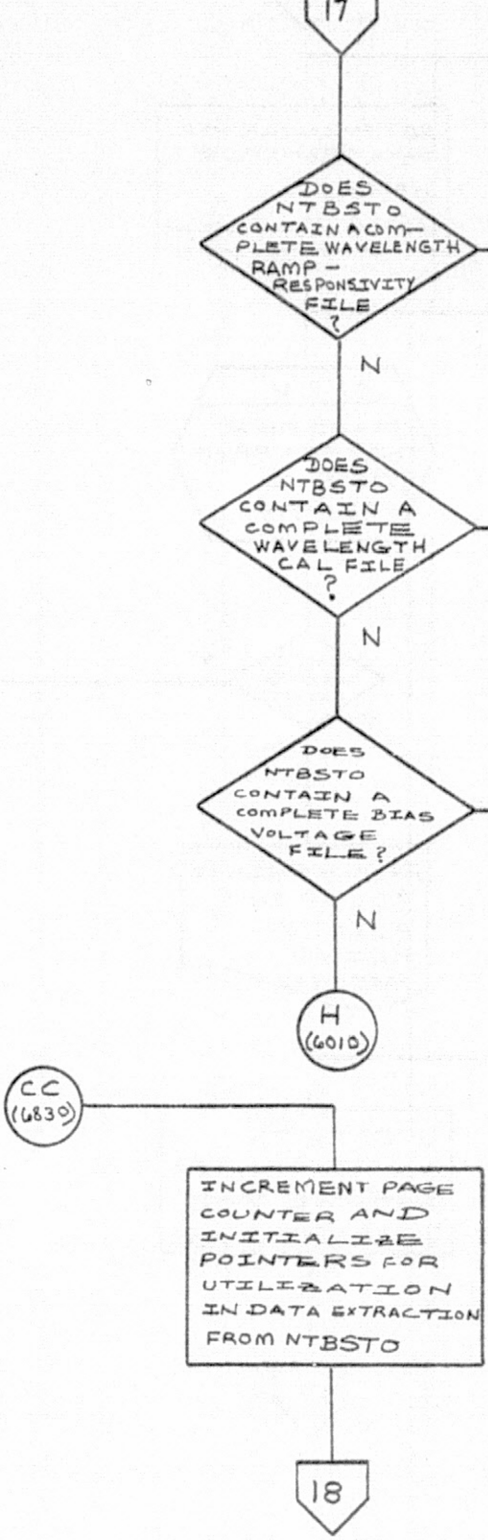

13
DEBUG PRINT: PDUMP THE DATE FROM THE RECORD BEING PROCESSED
W (6700)
V (6600)
HAVE NEW FILES BEEN INSERTED
Y
X (6705)
N
AMOV
TRANSFER THE RECORD JUST READ FROM NTABUF TO NTEMP
J (7000)
W (6700)
IS THE NTEMP ARRAY ZERO FILLED?
Y
X (6705)
N
14

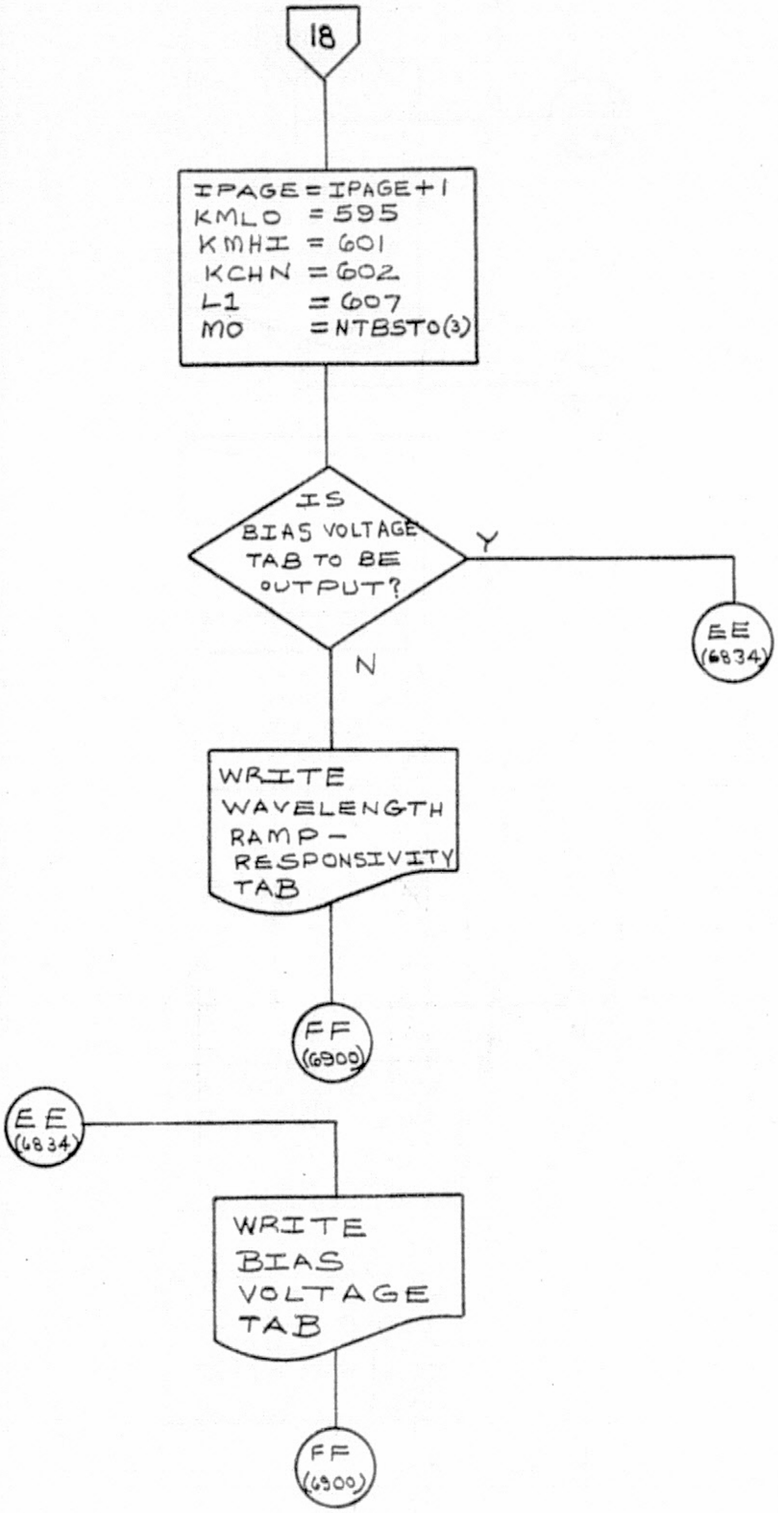

14
AMOV
TRANSFER THE DATA FROM NTEMP TO NTABUF
ZERO
CLEAR THE NTEMP ARRAY
X (6705)
NTRAN
WRITE A RECORD TO THE NEW HISTORICAL FILE TAPE
NTRAN
CHECK STATUS OF THE UNIT
IS TAB OF NEW FILE ONLY REQUIRED ?
Y
H (6010)
N
15

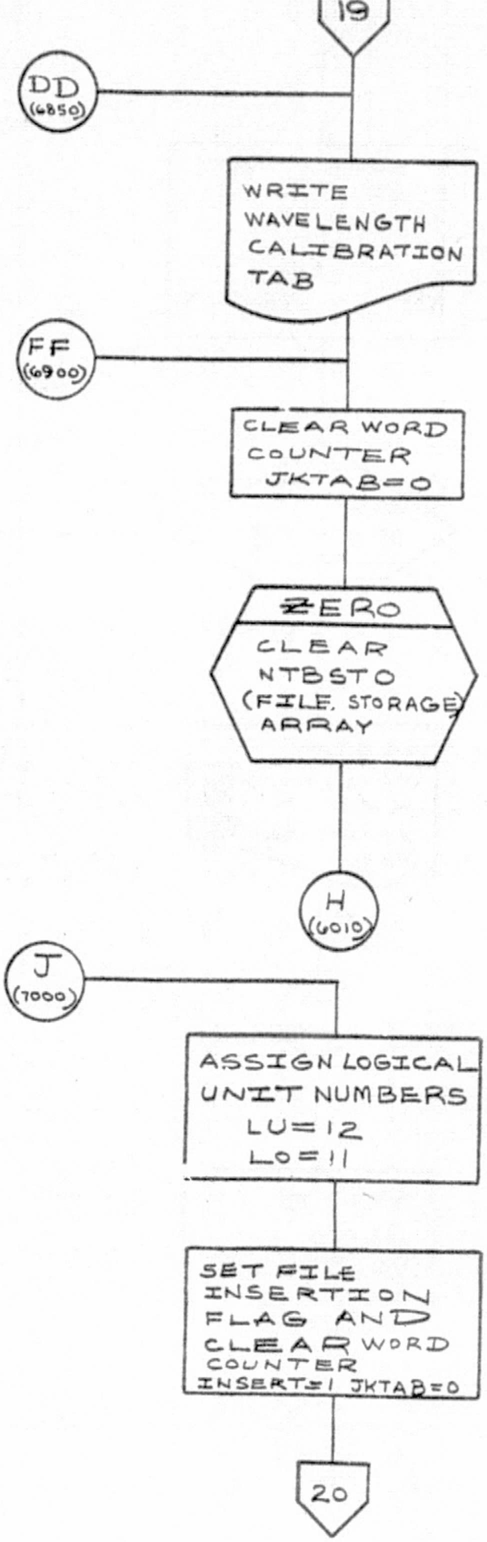

15
IS THE RECORD BEING PROCESSED OTHER THAN THE FIRST RECORD OF THE BIAS VOLTAGE FILE ?
Y
N
Y (6745)
SET POINTER FOR DATA TRANSFER
KMTAB=1
Z (6780)
Y (6745)
IS THE RECORD BEING PROCESSED A WAVE LENGTH CAL FILE RECORD ?
Y
N
AA (6750)
SET POINTER FOR DATA TRANSFER
KMTAB =
((NTABUF(1)-1) * 50)+1
Z (6780)

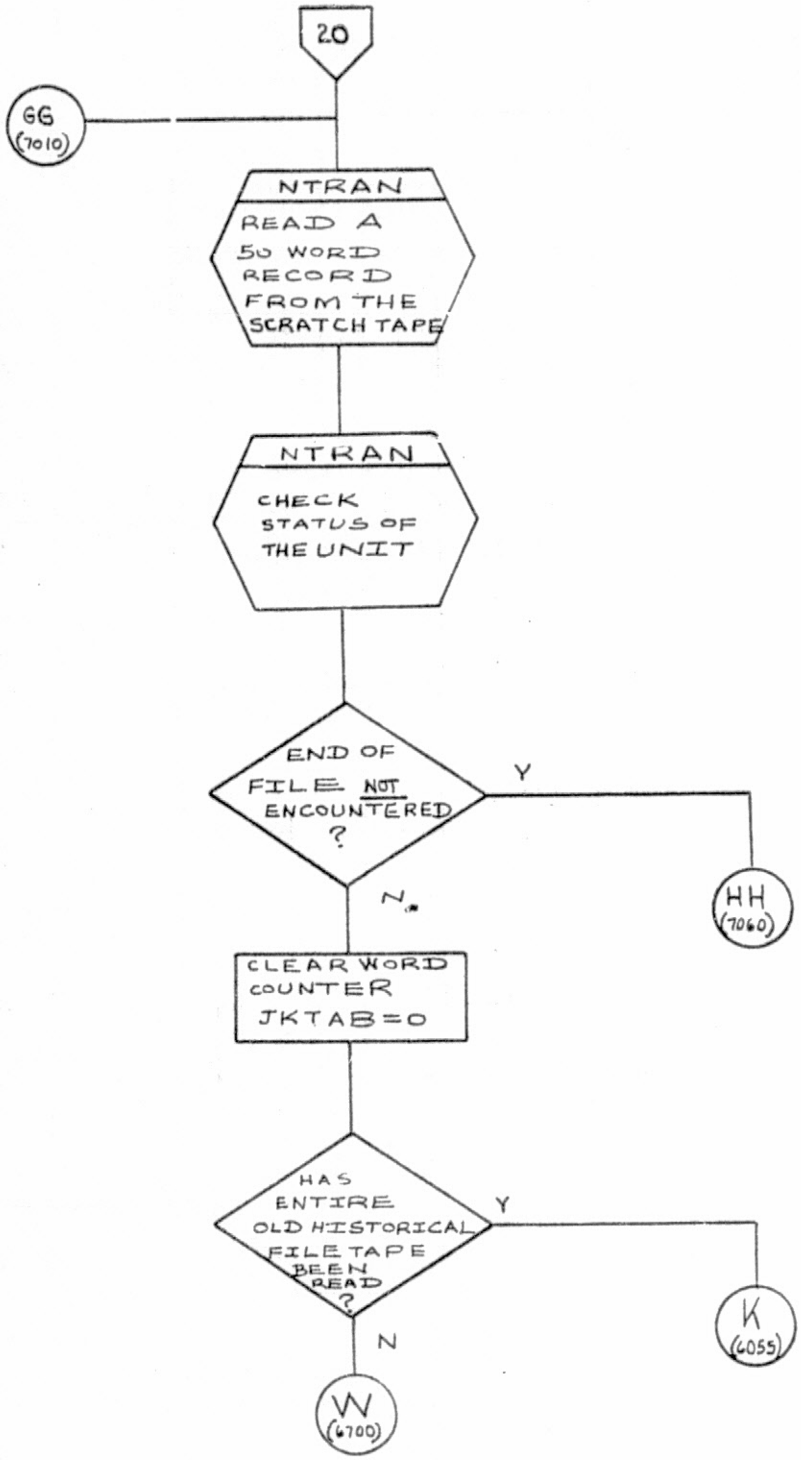


16
AA (6750)
SET POINTER FOR DATA TRANSFER KMTAB = ((NTABUF(1)-19)*50)+1
Z (6780)
AMOV
TRANSFER THE DATA FROM NTABUF TO NTBSTO
OMIT DEBUG PRINT ?
Y
N
BB (6790)
DEBUG PRINT: PDUMP THE RECORD BEING PROCESSED
BB (6790)
UPDATE THE WORD COUNTER FOR THE TAB JKTAB = JKTAB + 50
17

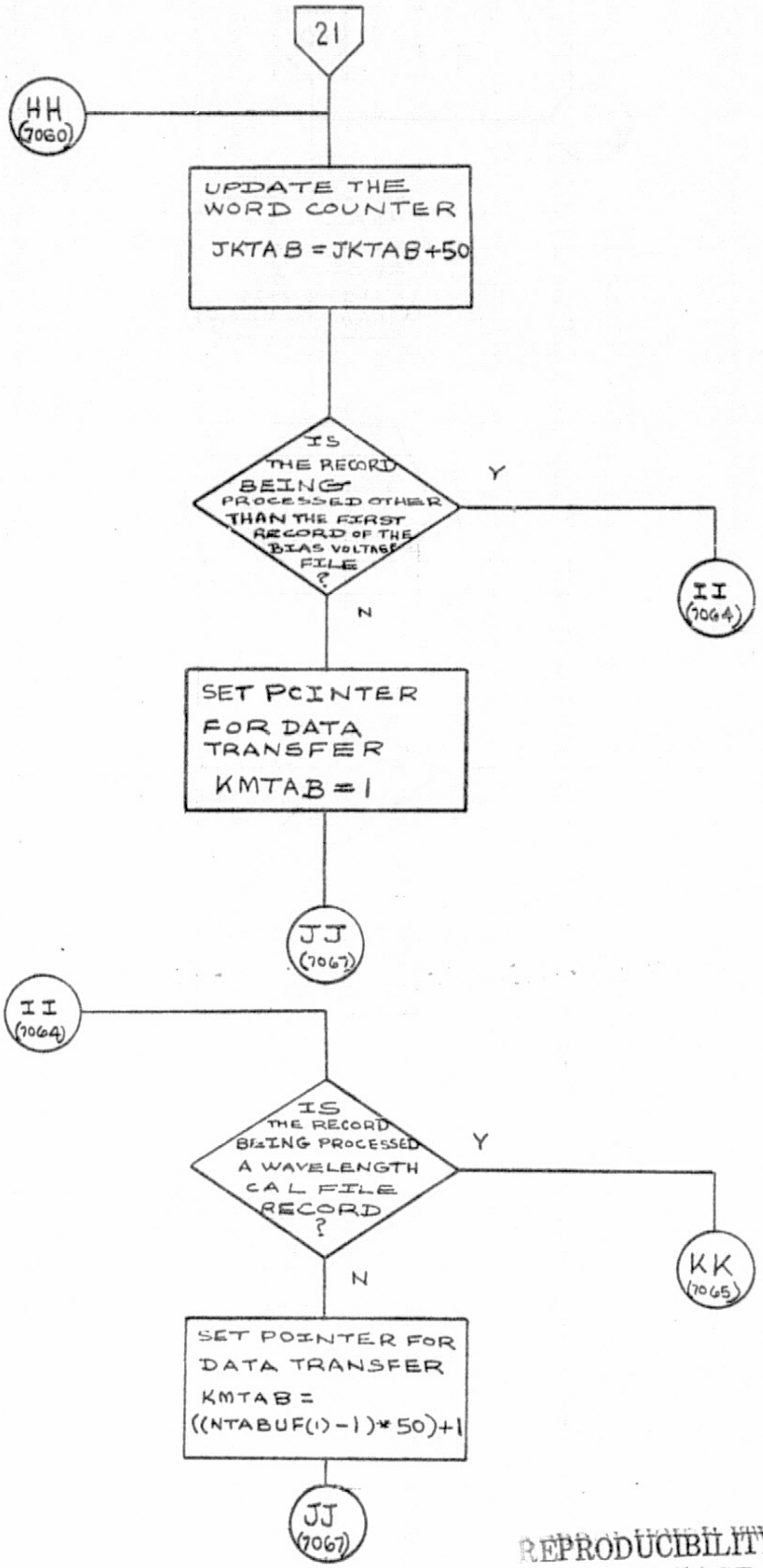

17
DOES NTBSTO CONTAIN A COMPLETE WAVELENGTH RAMP - RESPONSIVITY FILE ?
Y
N
CC (6830)
DOES NTBSTO CONTAIN A COMPLETE WAVELENGTH CAL FILE ?
Y
N
DD (6850)
DOES NTBSTO CONTAIN A COMPLETE BIAS VOLTAGE FILE ?
Y
N
CC (6830)
H (6010)
CC (6830)
INCREMENT PAGE COUNTER AND INITIALIZE POINTERS FOR UTILIZATION IN DATA EXTRACTION FROM NTBSTO
18

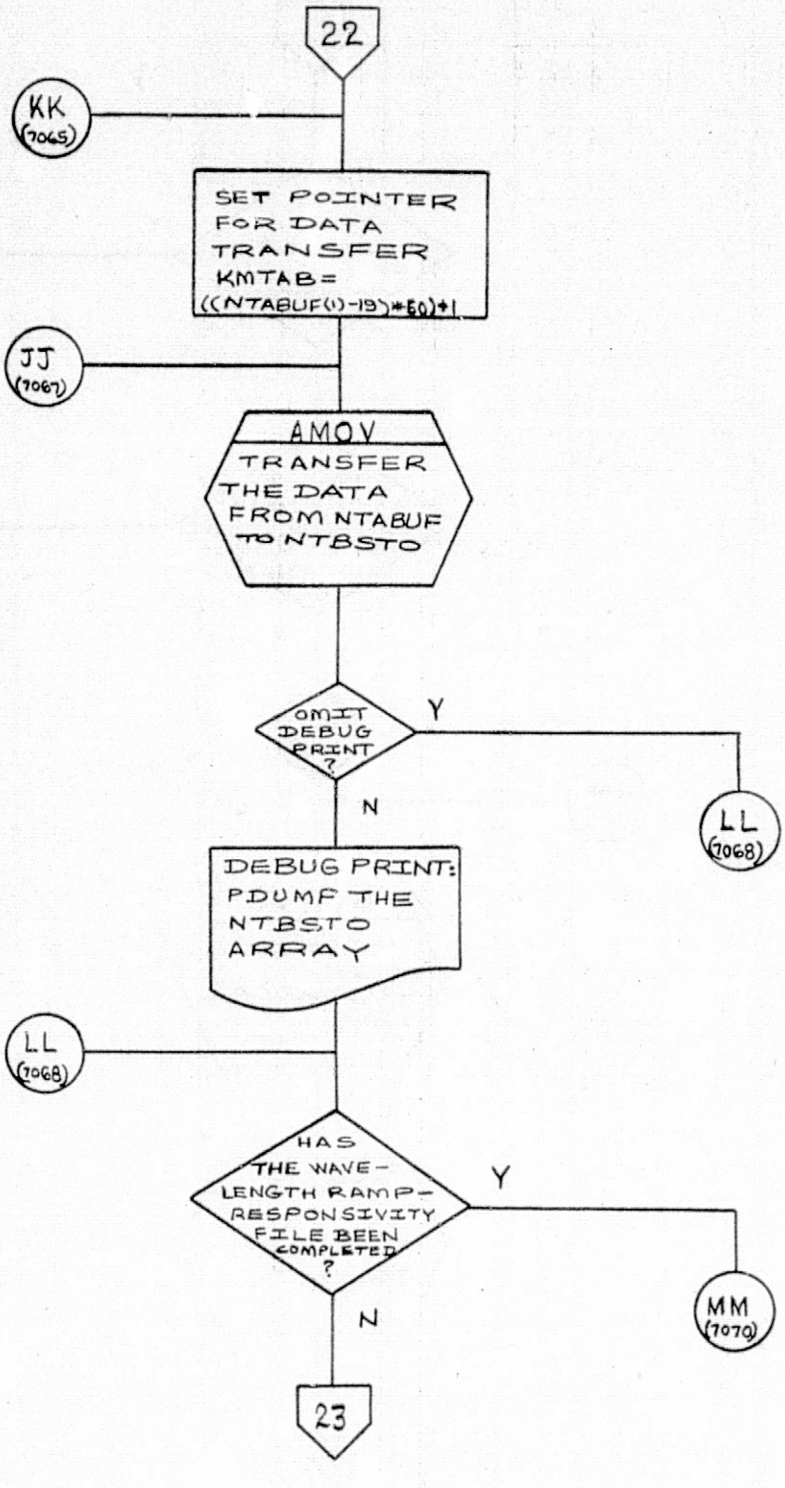

18
IPAGE = IPAGE + 1
KMLO = 595
KMHI = 601
KCHN = 602
L1 = 607
MO = NTBSTO(3)
IS BIAS VOLTAGE TAB TO BE OUTPUT?
Y
N
EE (6834)
WRITE WAVELENGTH RAMP – RESPONSIVITY TAB
FF (6900)
EE (6834)
WRITE BIAS VOLTAGE TAB
FF (6900)

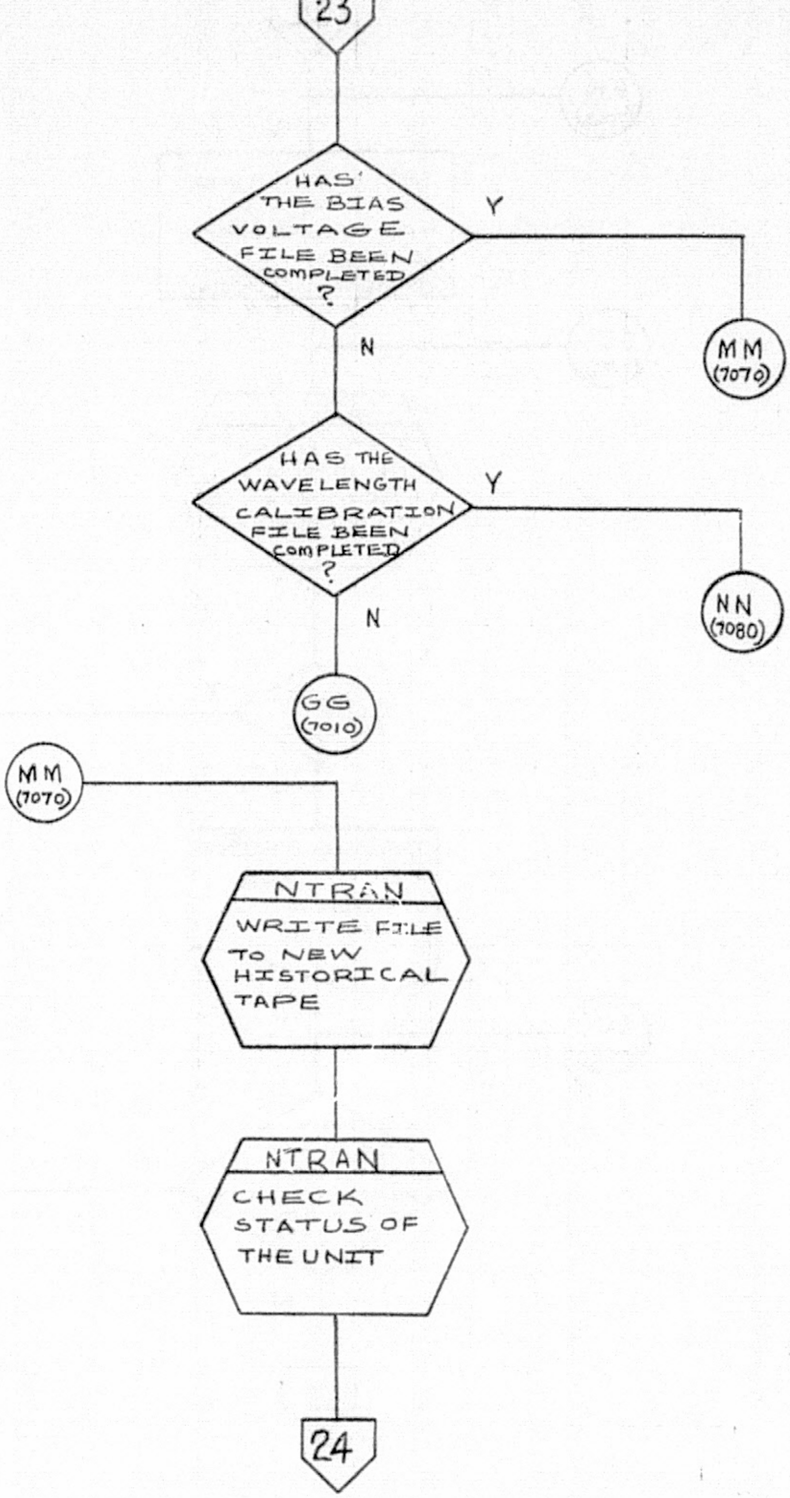

19
DD
(6850)
WRITE
WAVELENGTH
CALIBRATION
TAB
FF
(6900)
CLEAR WORD
COUNTER
JKTAB=0
ZERO
CLEAR
NTBSTO
(FILE STORAGE)
ARRAY
H
(6010)
J
(7000)
ASSIGN LOGICAL
UNIT NUMBERS
LU=12
LO=11
SET FILE
INSERTION
FLAG AND
CLEAR WORD
COUNTER
INSERT=1 JKTAB=0
20

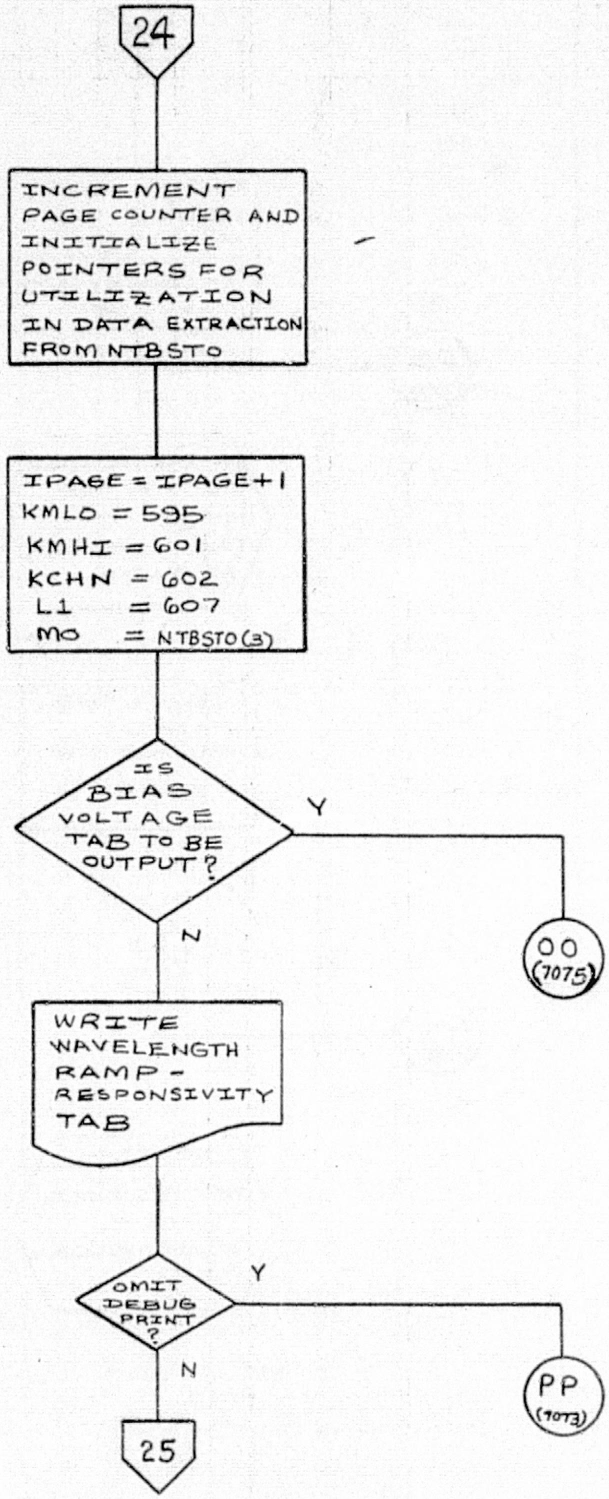


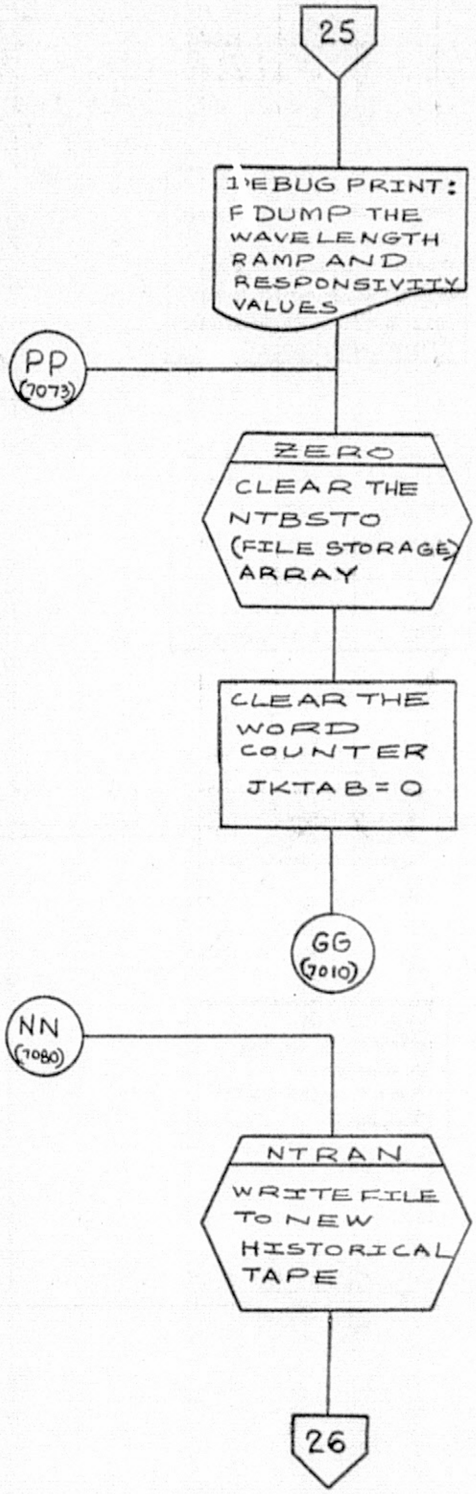
20
GG (7010)
NTRAN
READ A 50 WORD RECORD FROM THE SCRATCH TAPE
NTRAN
CHECK STATUS OF THE UNIT
END OF FILE NOT ENCOUNTERED ?
Y
HH (7060)
N
CLEAR WORD COUNTER JKTAB=0
HAS ENTIRE OLD HISTORICAL FILE TAPE BEEN READ ?
Y
K (6055)
N
W (6700)

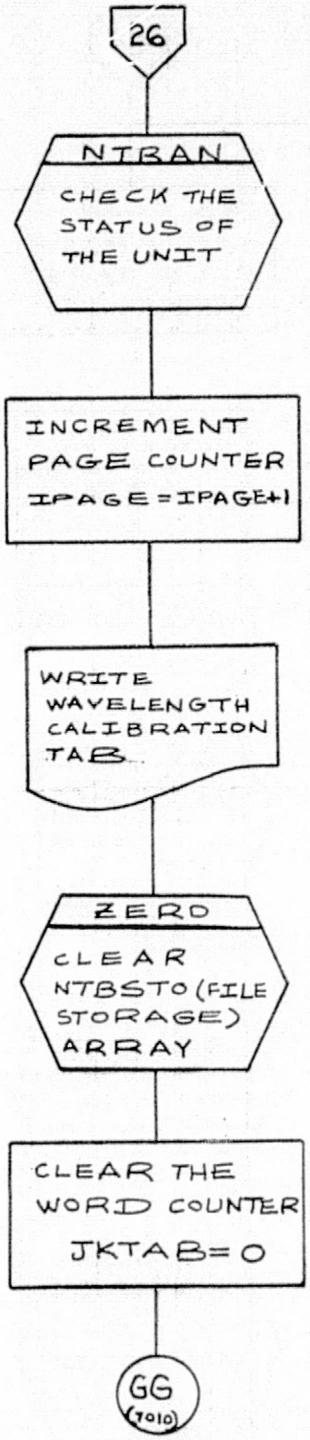

21
HH (7060)
UPDATE THE WORD COUNTER
JKTAB = JKTAB+50
IS THE RECORD BEING PROCESSED OTHER THAN THE FIRST RECORD OF THE BIAS VOLTAGE FILE ?
Y
II (7064)
N
SET POINTER FOR DATA TRANSFER
KMTAB = 1
JJ (7067)
II (7064)
IS THE RECORD BEING PROCESSED A WAVELENGTH CAL FILE RECORD ?
Y
KK (7065)
N
SET POINTER FOR DATA TRANSFER
KMTAB = ((NTABUF(1) - 1) * 50) + 1
JJ (7067)

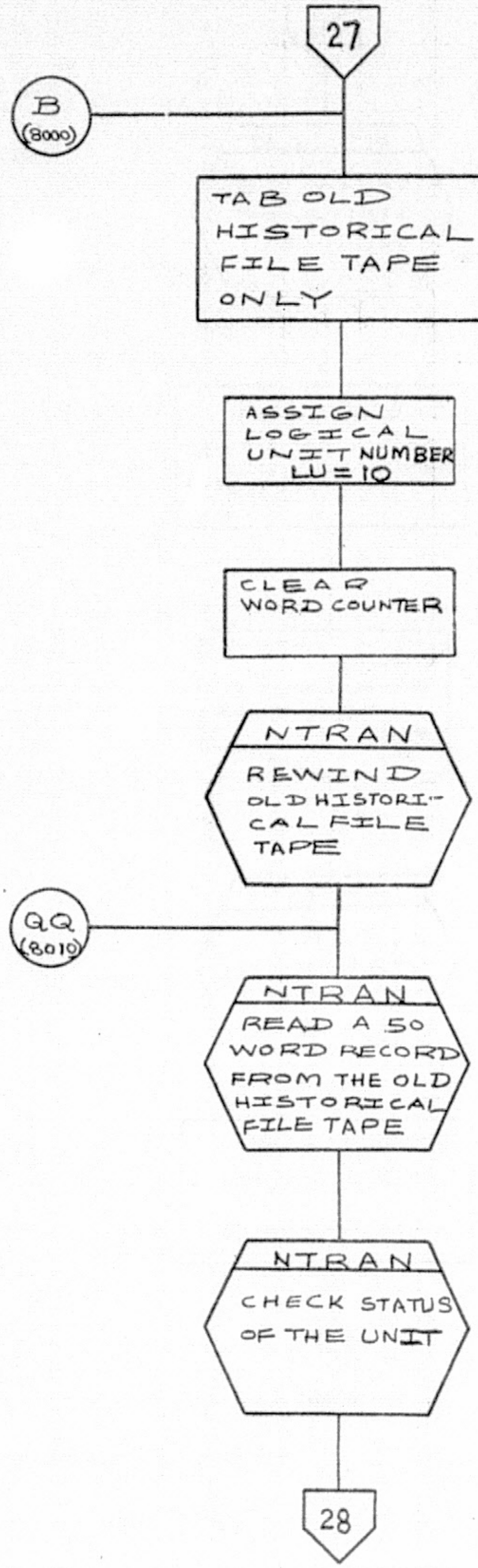

22
KK
(7065)
SET POINTER FOR DATA TRANSFER KMTAB= ((NTABUF(1)-19)*60)+1
JJ
(7067)
AMOV
TRANSFER THE DATA FROM NTABUF TO NTBSTO
OMIT DEBUG PRINT ?
Y
N
LL
(7068)
DEBUG PRINT: PDUMP THE NTBSTO ARRAY
LL
(7068)
HAS THE WAVE-LENGTH RAMP-RESPONSIVITY FILE BEEN COMPLETED ?
Y
N
MM
(7070)
23

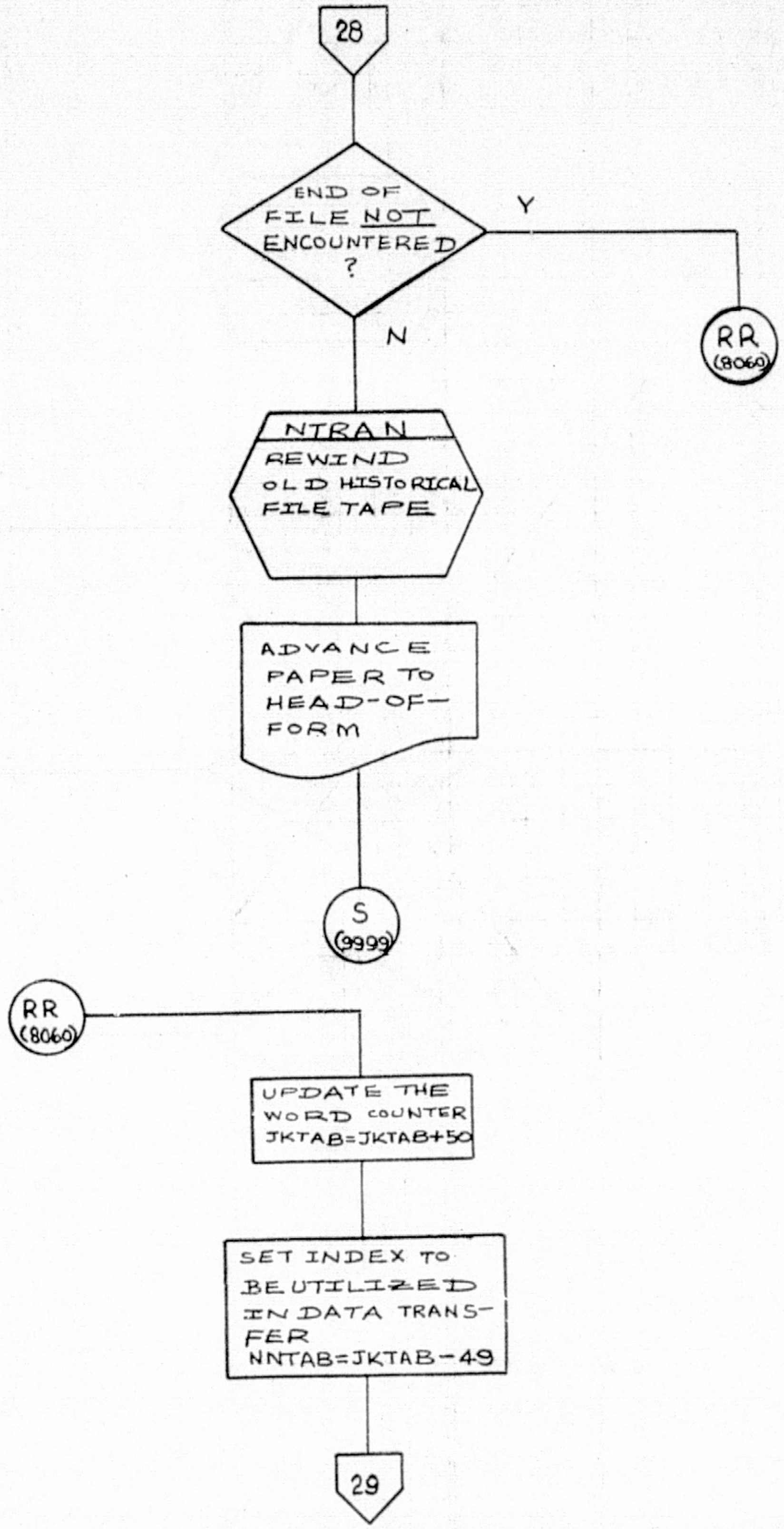

23
HAS THE BIAS VOLTAGE FILE BEEN COMPLETED ?
Y
N
MM (7070)
HAS THE WAVELENGTH CALIBRATION FILE BEEN COMPLETED ?
Y
N
NN (7080)
GG (7010)
MM (7070)
NTRAN
WRITE FILE TO NEW HISTORICAL TAPE
NTRAN
CHECK STATUS OF THE UNIT
24

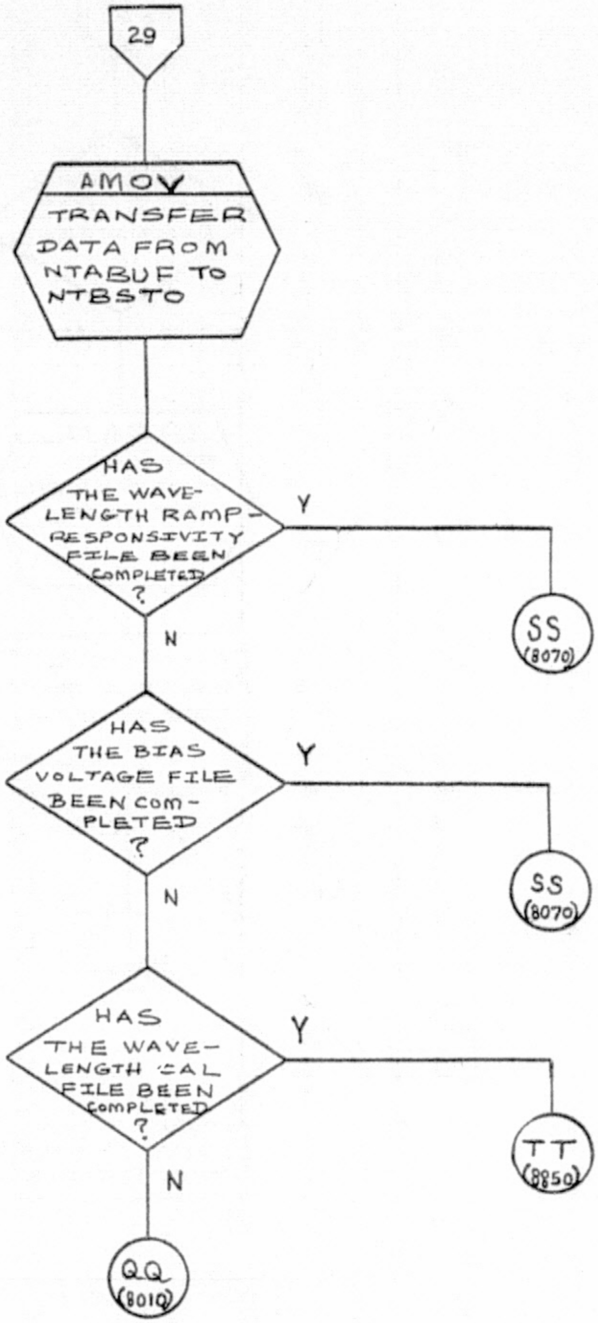

24
INCREMENT PAGE COUNTER AND INITIALIZE POINTERS FOR UTILIZATION IN DATA EXTRACTION FROM NTBSTO
IPAGE = IPAGE+1
KMLO = 595
KMHI = 601
KCHN = 602
L1 = 607
MO = NTBSTO(3)
IS BIAS VOLTAGE TAB TO BE OUTPUT?
Y
N
OO (7075)
WRITE WAVELENGTH RAMP - RESPONSIVITY TAB
OMIT DEBUG PRINT ?
Y
N
PP (7073)
25

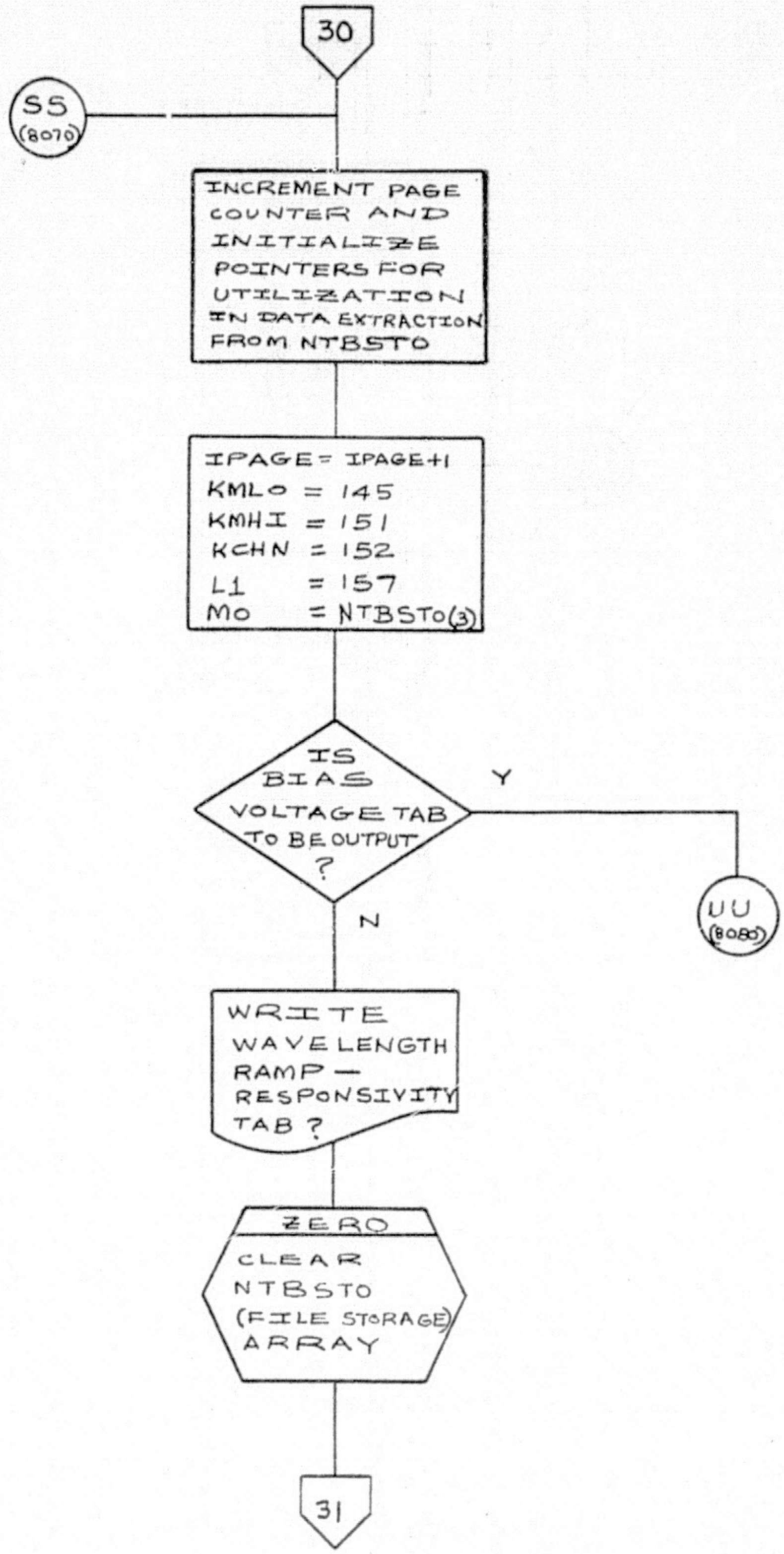

25
DEBUG PRINT: F DUMP THE WAVELENGTH RAMP AND RESPONSIVITY VALUES
PP (7073)
ZERO
CLEAR THE NTBSTO (FILE STORAGE) ARRAY
CLEAR THE WORD COUNTER JKTAB=0
GG (7010)
NN (7080)
NTRAN
WRITE FILE TO NEW HISTORICAL TAPE
26

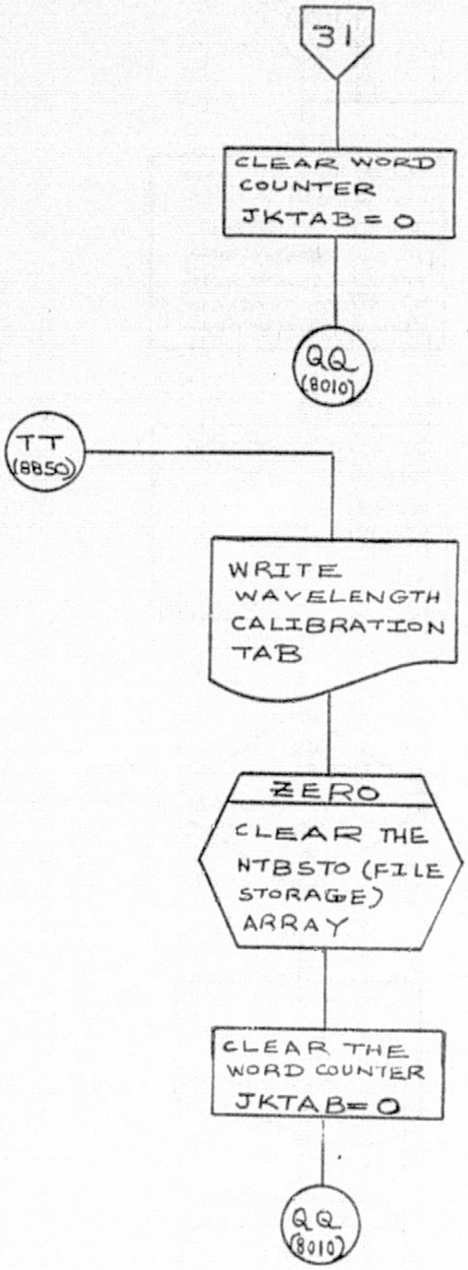

26
NTRAN
CHECK THE STATUS OF THE UNIT
INCREMENT PAGE COUNTER
IPAGE=IPAGE+1
WRITE WAVELENGTH CALIBRATION TAB
ZERO
CLEAR NTBSTO (FILE STORAGE) ARRAY
CLEAR THE WORD COUNTER
JKTAB=0
GG
(7010)

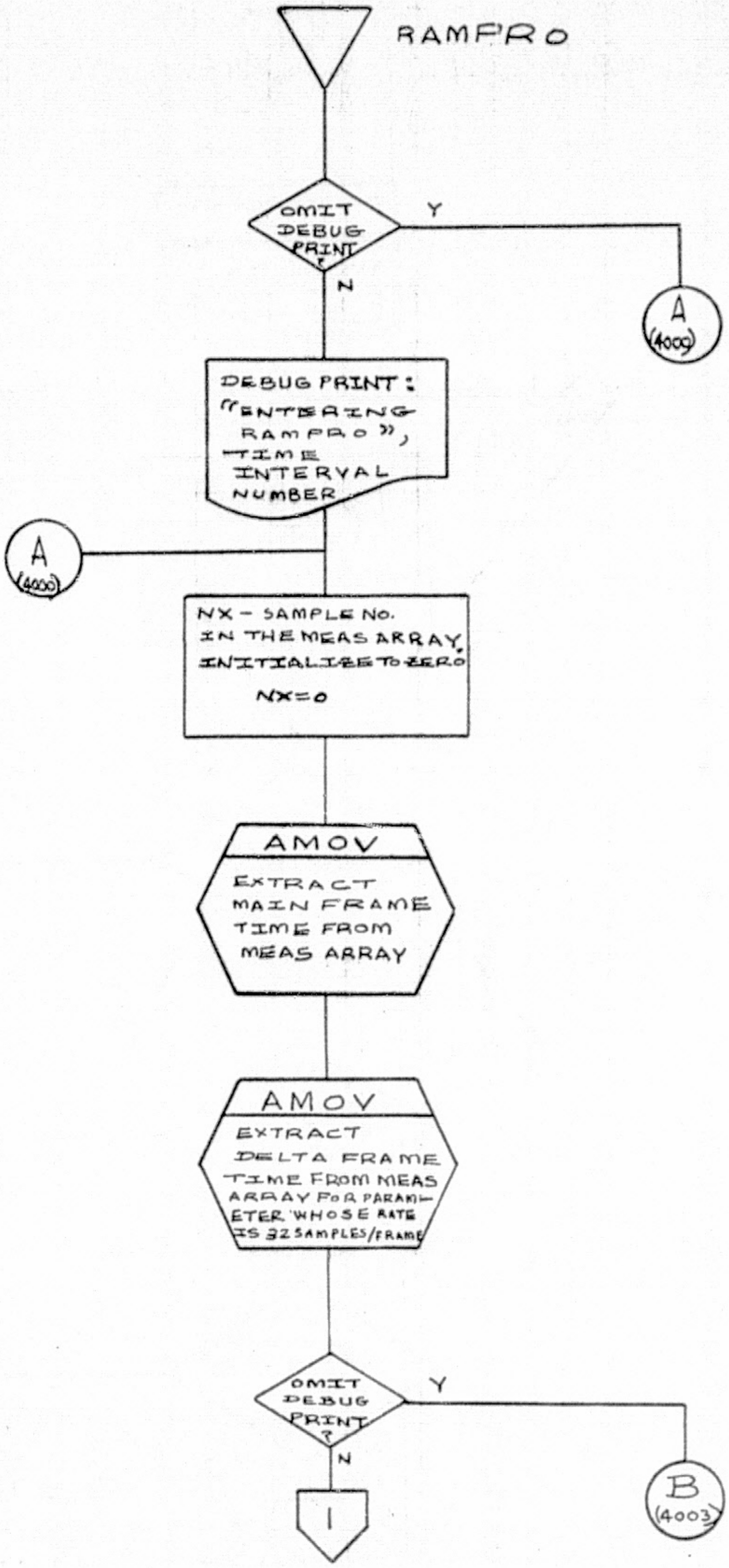

27
B
(8000)
TAB OLD HISTORICAL FILE TAPE ONLY
ASSIGN LOGICAL UNIT NUMBER LU=10
CLEAR WORD COUNTER
NTRAN
REWIND OLD HISTORI-CAL FILE TAPE
QQ
(8010)
NTRAN
READ A 50 WORD RECORD FROM THE OLD HISTORICAL FILE TAPE
NTRAN
CHECK STATUS OF THE UNIT
28

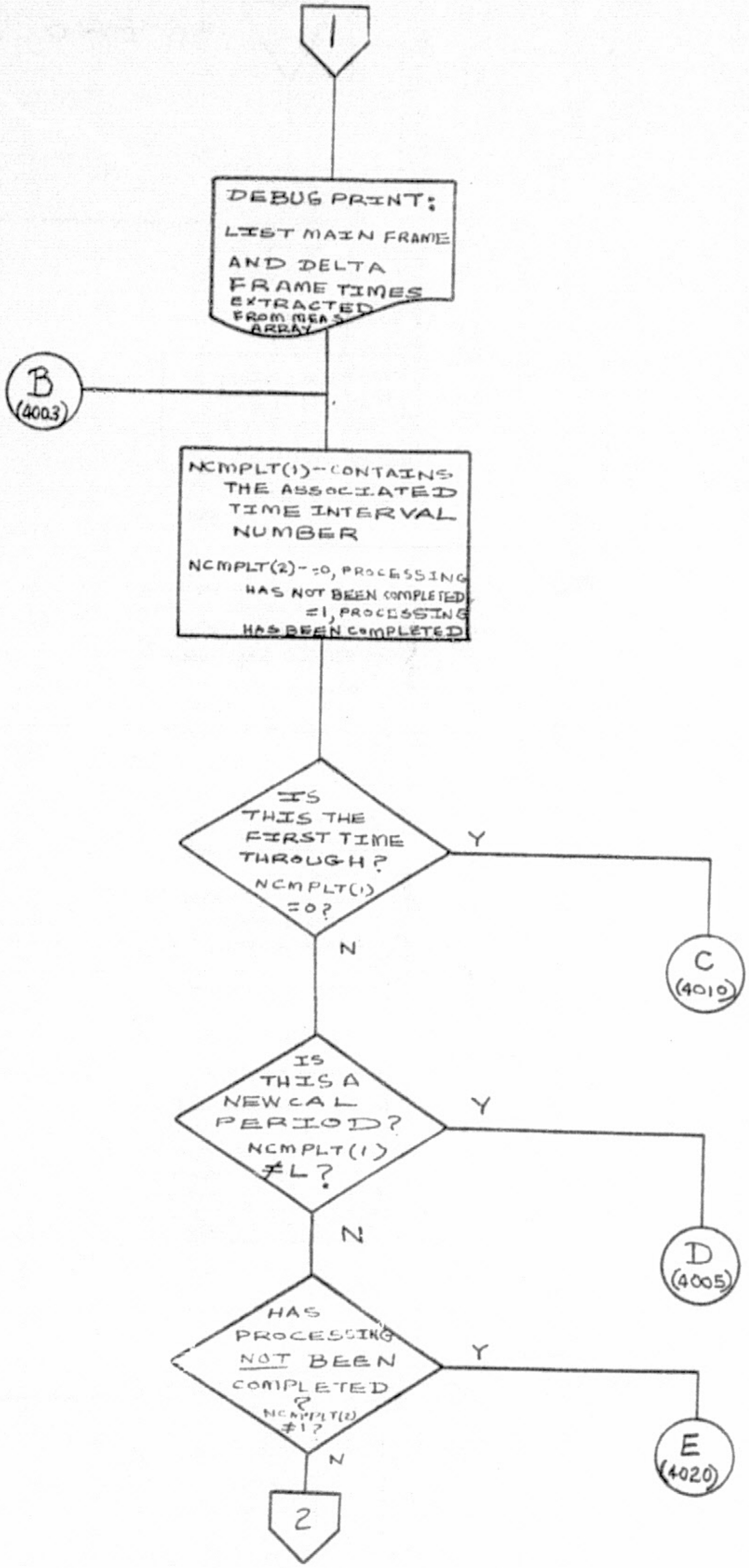

28
END OF FILE NOT ENCOUNTERED ?
Y
N
RR (8060)
NTRAN
REWIND OLD HISTORICAL FILE TAPE
ADVANCE PAPER TO HEAD-OF-FORM
S (9999)
RR (8060)
UPDATE THE WORD COUNTER JKTAB=JKTAB+50
SET INDEX TO BE UTILIZED IN DATA TRANSFER NNTAB=JKTAB-49
29

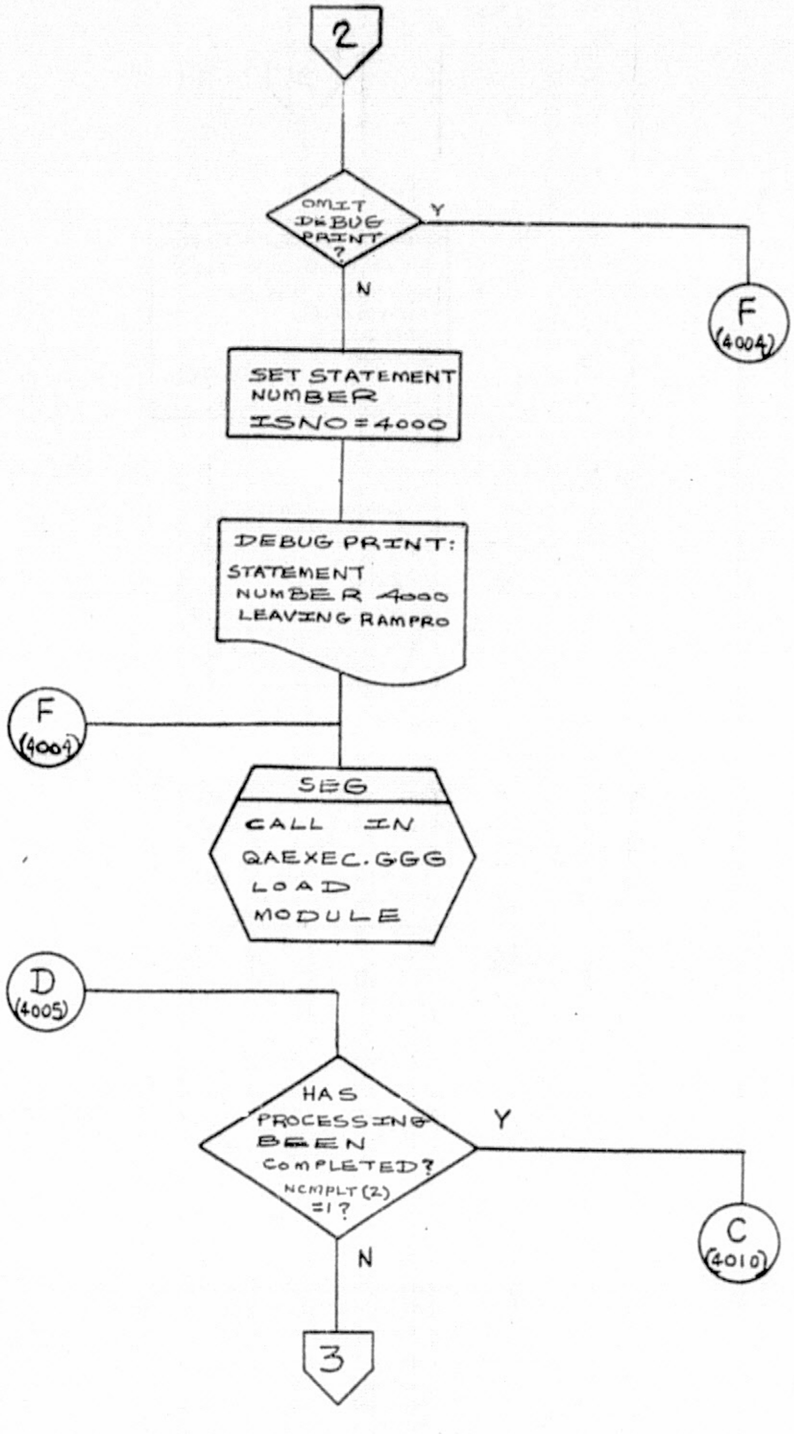

29
AMOV
TRANSFER DATA FROM NTABUF TO NTBSTO
HAS THE WAVE-LENGTH RAMP-RESPONSIVITY FILE BEEN COMPLETED ?
Y
SS (8070)
N
HAS THE BIAS VOLTAGE FILE BEEN COM-PLETED ?
Y
SS (8070)
N
HAS THE WAVE-LENGTH CAL FILE BEEN COMPLETED ?
Y
TT (8850)
N
QQ (8010)

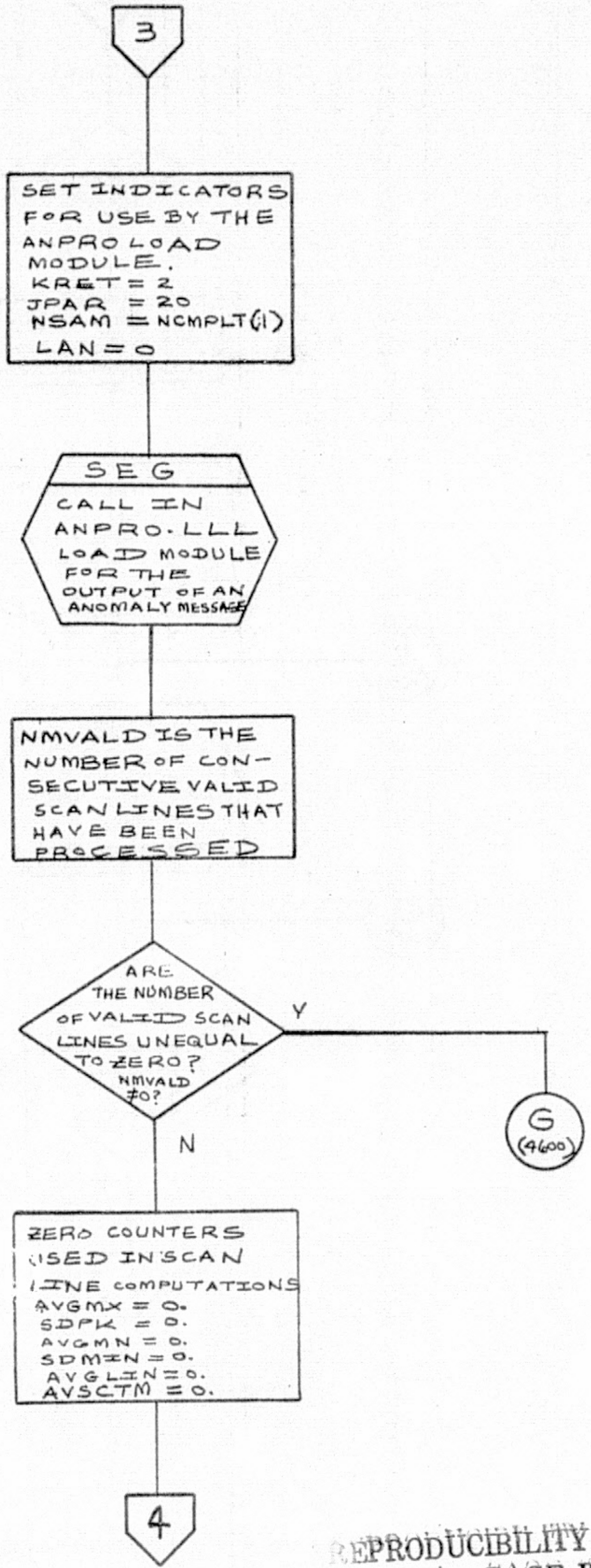

30
SS
(8070)
INCREMENT PAGE COUNTER AND INITIALIZE POINTERS FOR UTILIZATION IN DATA EXTRACTION FROM NTBSTO
IPAGE = IPAGE+1
KMLO = 145
KMHI = 151
KCHN = 152
L1 = 157
MO = NTBSTO(3)
IS BIAS VOLTAGE TAB TO BE OUTPUT ?
Y
UU
(8080)
N
WRITE WAVELENGTH RAMP — RESPONSIVITY TAB ?
ZERO
CLEAR NTBSTO (FILE STORAGE) ARRAY
31

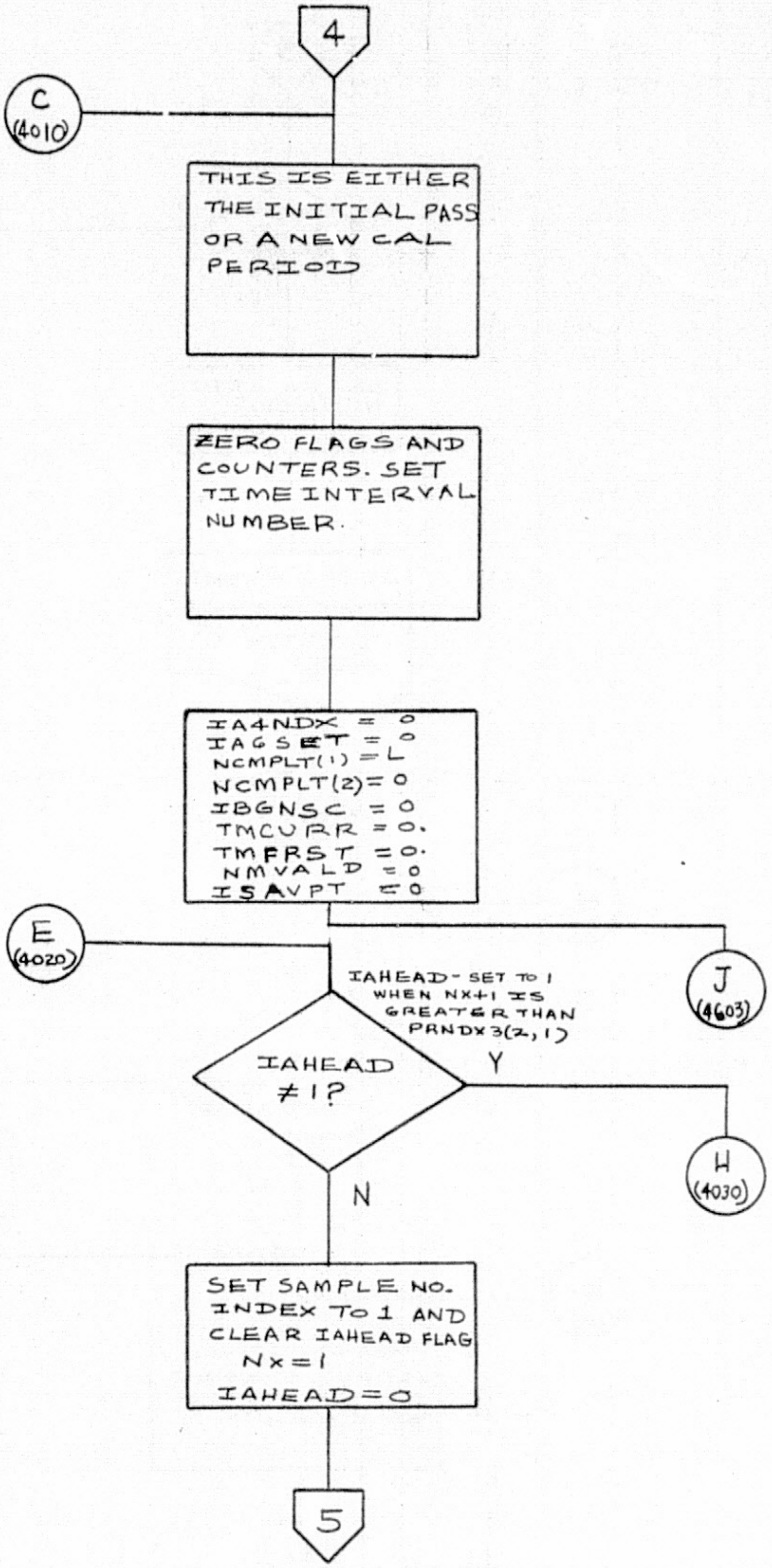

31
CLEAR WORD COUNTER JKTAB=0
QQ (8010)
TT (8850)
WRITE WAVELENGTH CALIBRATION TAB
ZERO
CLEAR THE NTBSTO (FILE STORAGE) ARRAY
CLEAR THE WORD COUNTER JKTAB=0
QQ (8010)

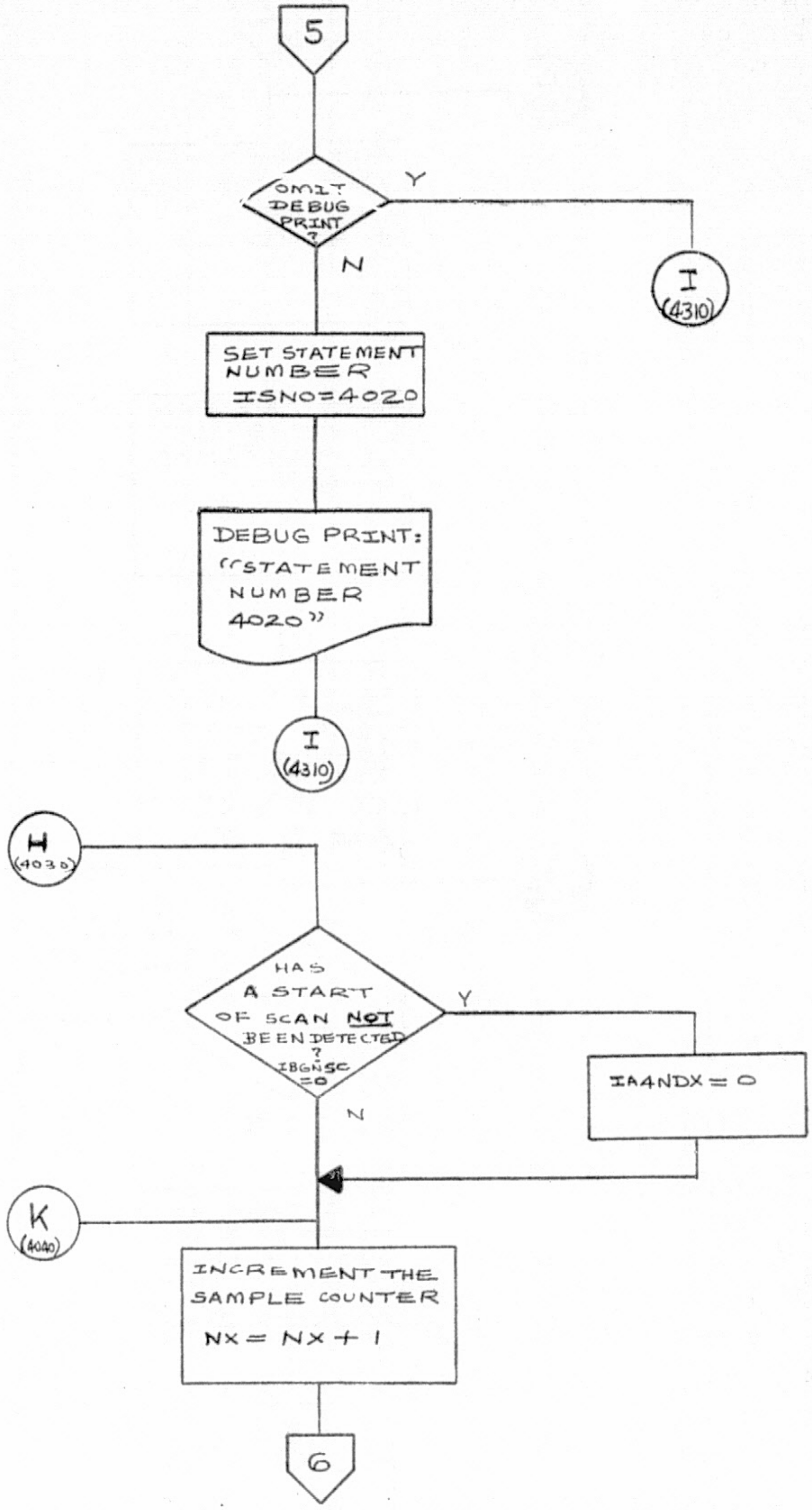

RAMFRO
OMIT DEBUG PRINT ?
Y
N
A (4000)
DEBUG PRINT: "ENTERING RAMFRO", TIME INTERVAL NUMBER
A (4000)
NX - SAMPLE NO. IN THE MEAS ARRAY, INITIALIZE TO ZERO
NX=0
AMOV
EXTRACT MAIN FRAME TIME FROM MEAS ARRAY
AMOV
EXTRACT DELTA FRAME TIME FROM MEAS ARRAY FOR PARAMETER WHOSE RATE IS 32 SAMPLES/FRAME
OMIT DEBUG PRINT ?
Y
N
B (4003)
1

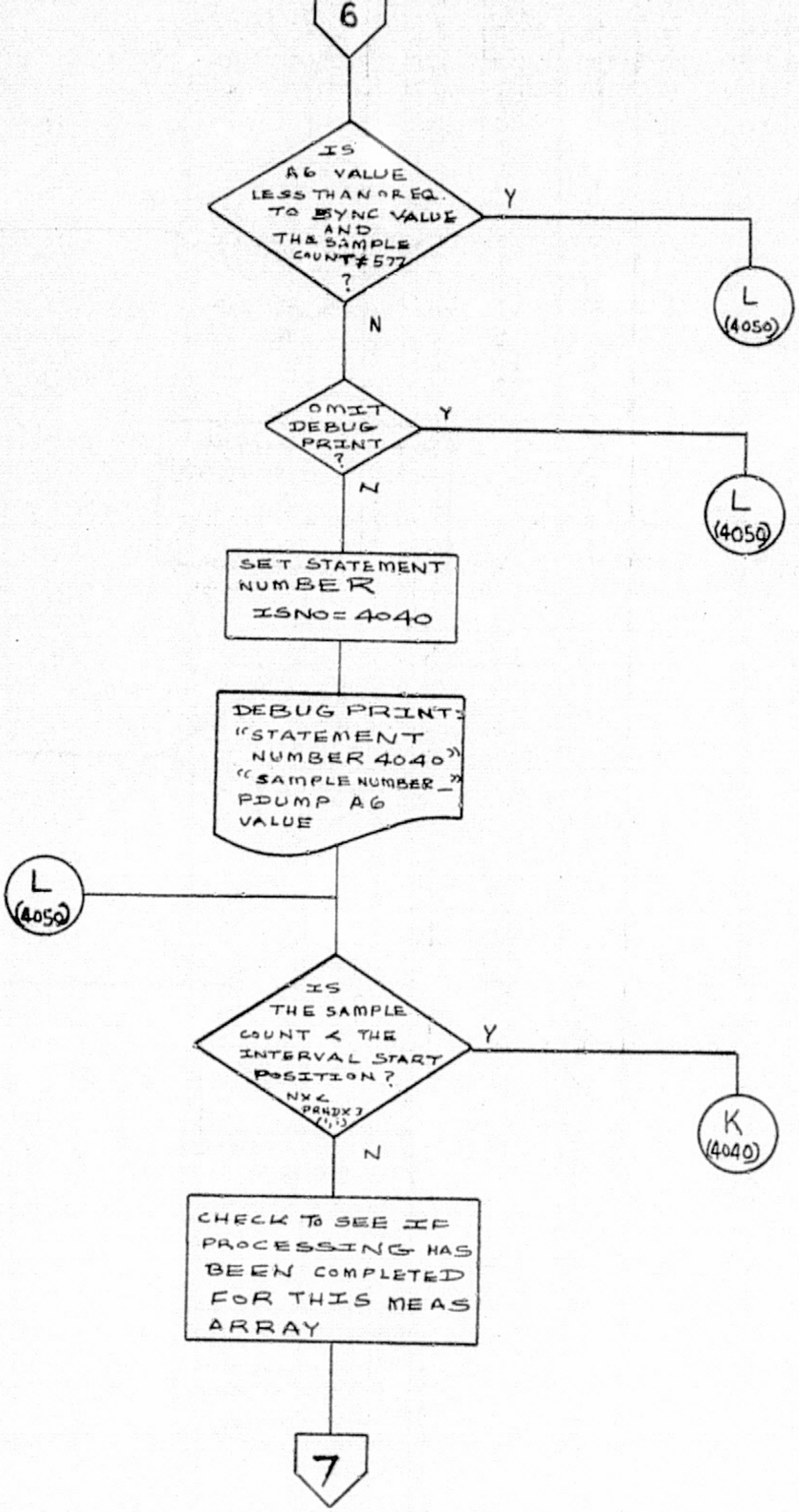

1
DEBUG PRINT:
LIST MAIN FRAME
AND DELTA
FRAME TIMES
EXTRACTED
FROM MEAS
ARRAY
B
(4003)
NCMPLT(1)—CONTAINS
THE ASSOCIATED
TIME INTERVAL
NUMBER
NCMPLT(2)—=0, PROCESSING
HAS NOT BEEN COMPLETED
=1, PROCESSING
HAS BEEN COMPLETED
IS
THIS THE
FIRST TIME
THROUGH?
NCMPLT(1)
=0?
Y
N
C
(4010)
IS
THIS A
NEW CAL
PERIOD?
NCMPLT(1)
≠L?
Y
N
D
(4005)
HAS
PROCESSING
NOT BEEN
COMPLETED
?
NCMPLT(2)
≠1?
Y
N
E
(4020)
2

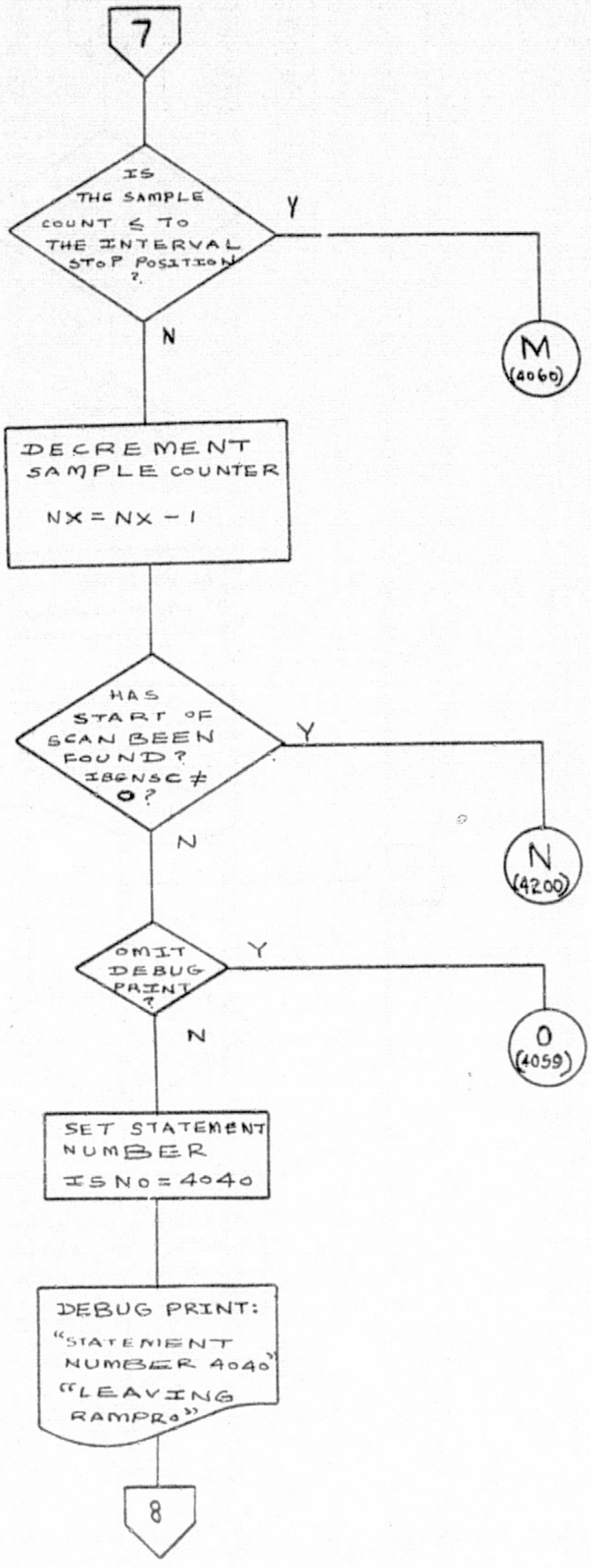

2
OMIT DEBUG PRINT ?
Y
F (4004)
N
SET STATEMENT NUMBER ISNO=4000
DEBUG PRINT: STATEMENT NUMBER 4000 LEAVING RAMPRO
F (4004)
SEG
CALL IN QAEXEC.GGG LOAD MODULE
D (4005)
HAS PROCESSING BEEN COMPLETED? NCMPLT(2) =1?
Y
C (4010)
N
3

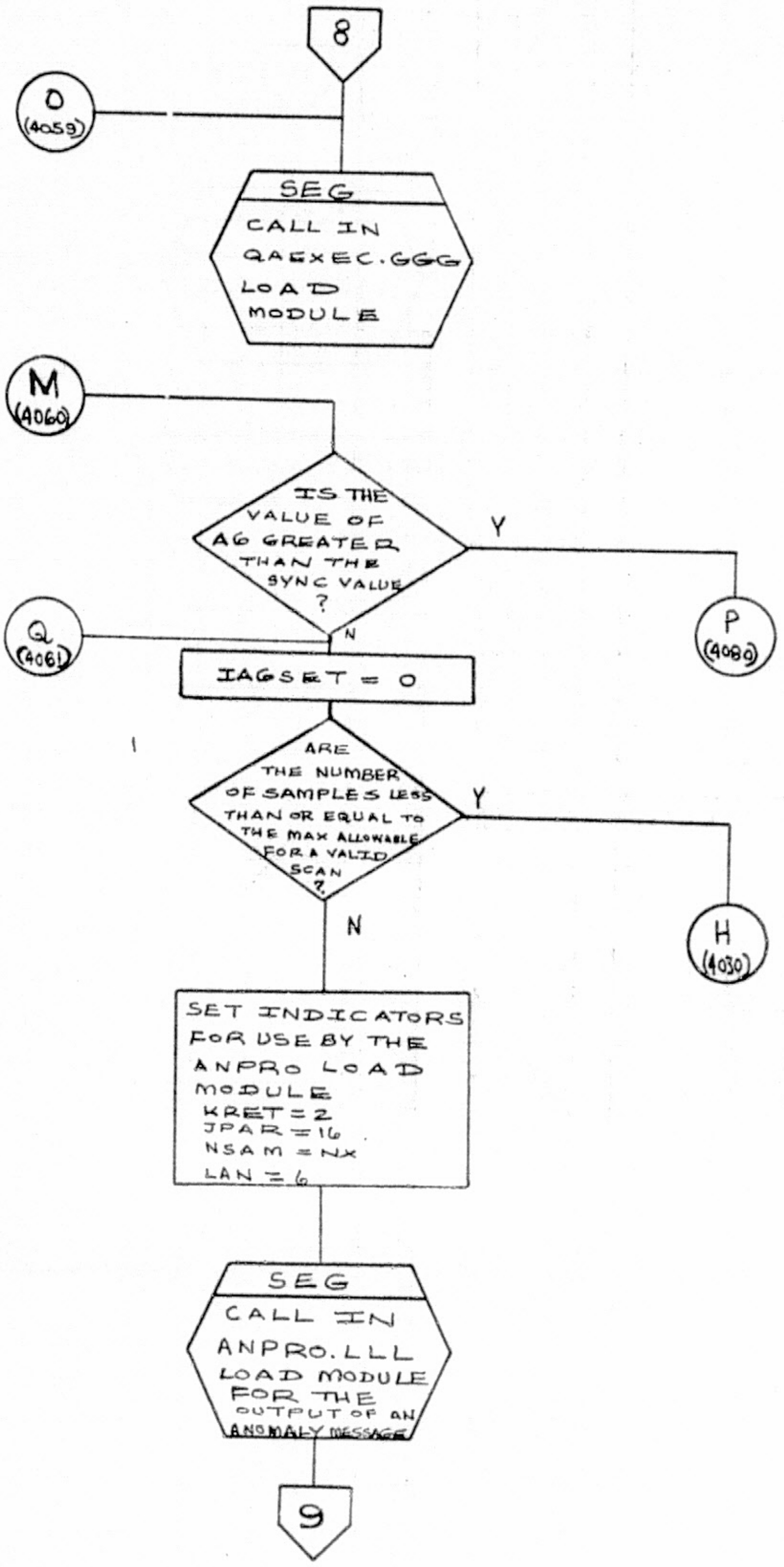

3
SET INDICATORS FOR USE BY THE ANPRO LOAD MODULE.
KRET = 2
JPAR = 20
NSAM = NCMPLT(1)
LAN = 0
SEG
CALL IN ANPRO.LLL LOAD MODULE FOR THE OUTPUT OF AN ANOMALY MESSAGE
NMVALD IS THE NUMBER OF CONSECUTIVE VALID SCAN LINES THAT HAVE BEEN PROCESSED
ARE THE NUMBER OF VALID SCAN LINES UNEQUAL TO ZERO? NMVALD ≠ 0?
Y
N
G (4600)
ZERO COUNTERS USED IN SCAN LINE COMPUTATIONS
AVGMX = 0.
SDPK = 0.
AVGMN = 0.
SDMIN = 0.
AVGLIN = 0.
AVSCTM = 0.
4

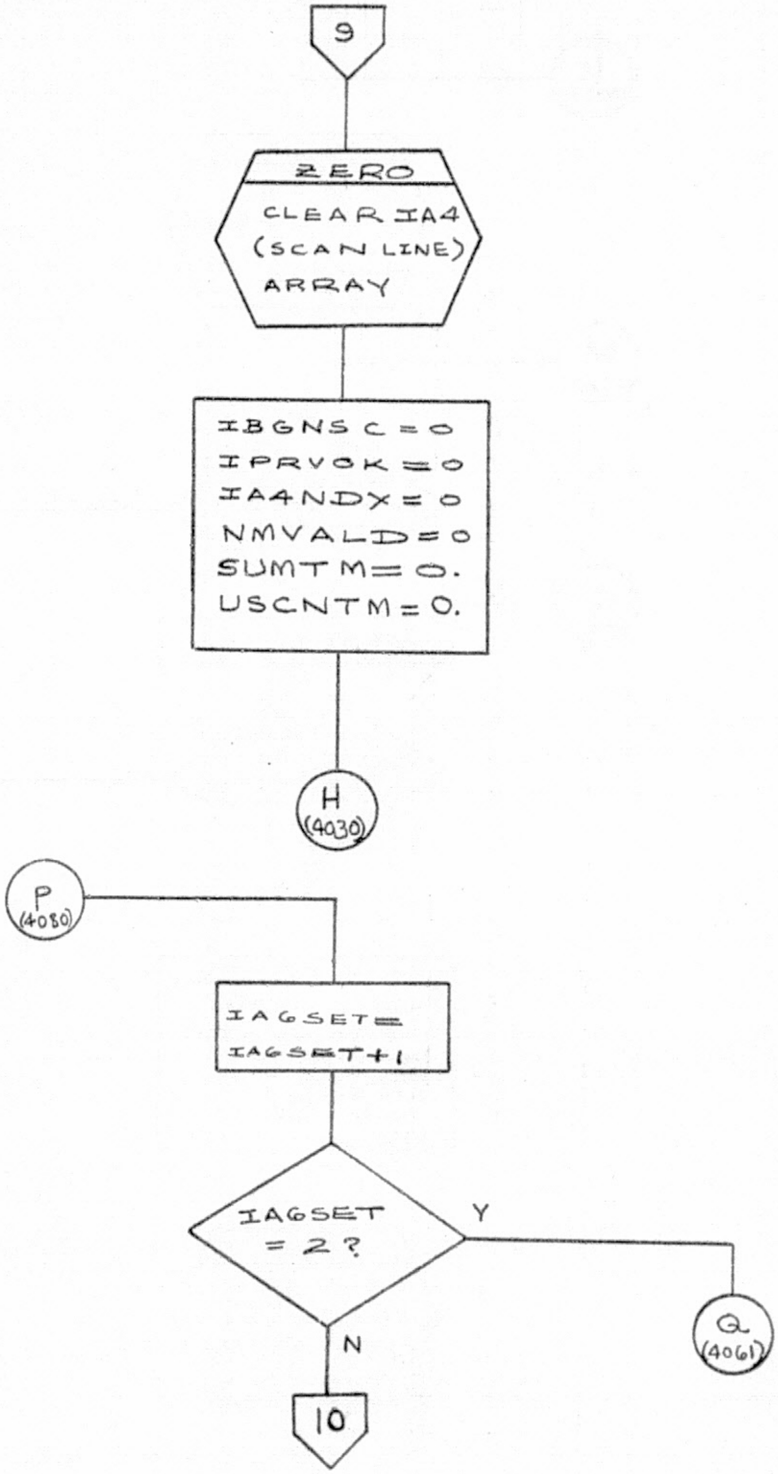

4
C
(4010)
THIS IS EITHER THE INITIAL PASS OR A NEW CAL PERIOD
ZERO FLAGS AND COUNTERS. SET TIME INTERVAL NUMBER.
IA4NDX = 0
IAGSET = 0
NCMPLT(1) = L
NCMPLT(2) = 0
IBGNSC = 0
TMCURR = 0.
TMFRST = 0.
NMVALD = 0
ISAVPT = 0
E
(4020)
J
(4603)
IAHEAD - SET TO 1 WHEN NX+1 IS GREATER THAN PRNDX3(2,1)
IAHEAD ≠ 1?
Y
H
(4030)
N
SET SAMPLE NO. INDEX TO 1 AND CLEAR IAHEAD FLAG
Nx = 1
IAHEAD = 0
5

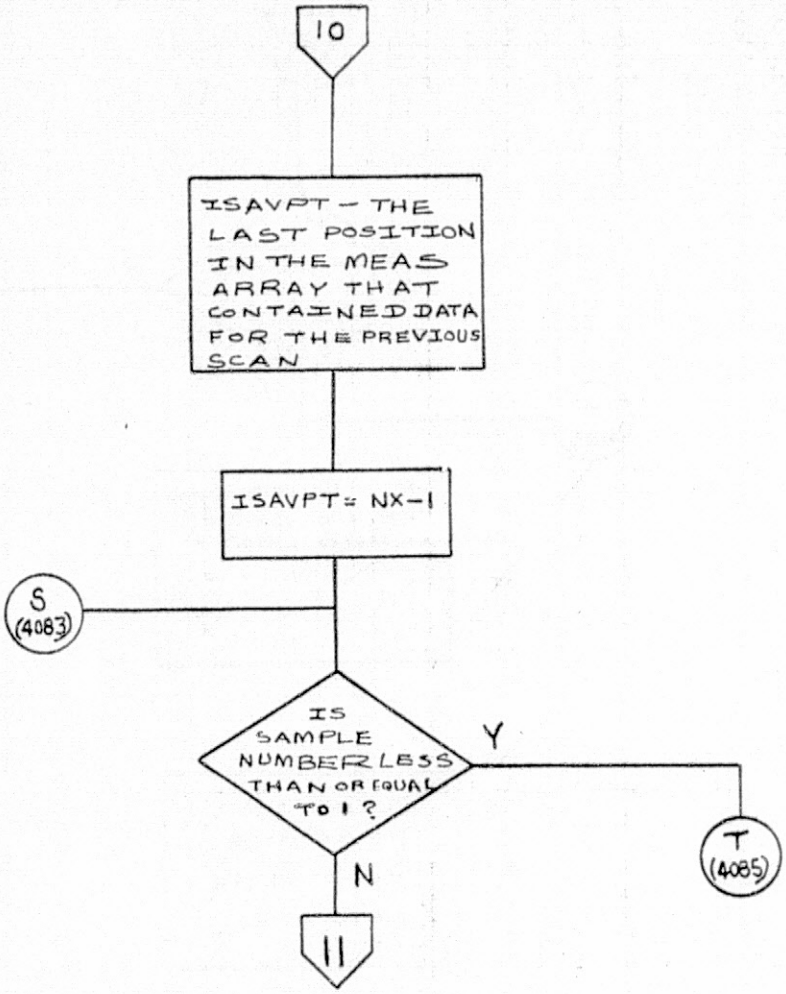

5
OMIT DEBUG PRINT ?
Y
N
I (4310)
SET STATEMENT NUMBER ISNO=4020
DEBUG PRINT: "STATEMENT NUMBER 4020"
I (4310)
H (4030)
HAS A START OF SCAN NOT BEEN DETECTED ? IBGNSC =0
Y
N
IA4NDX = 0
K (4040)
INCREMENT THE SAMPLE COUNTER NX = NX + 1
6

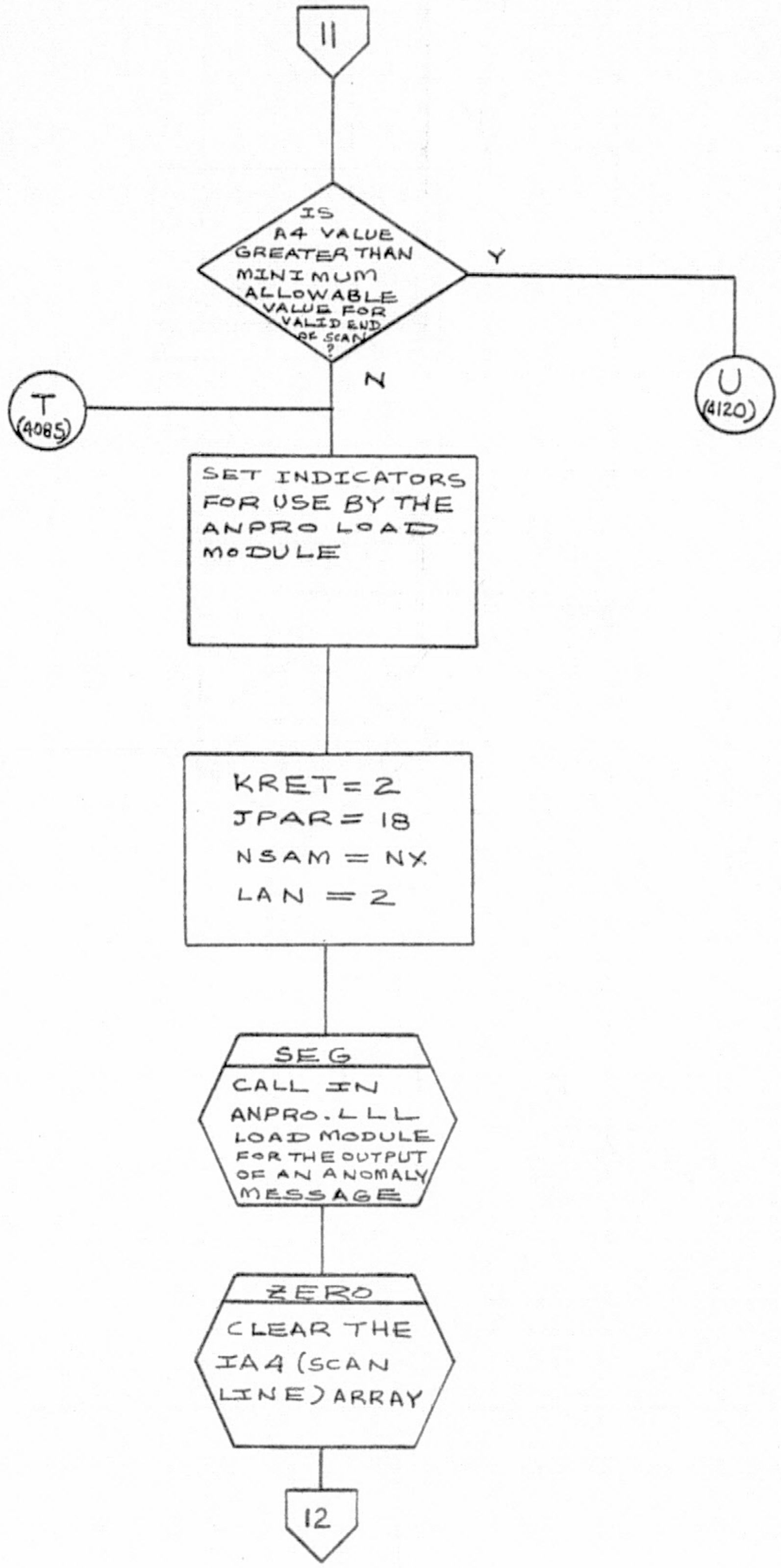

6
IS A6 VALUE LESS THAN OR EQ. TO SYNC VALUE AND THE SAMPLE COUNT ≠ 577 ?
Y
L (4050)
N
OMIT DEBUG PRINT ?
Y
L (4050)
N
SET STATEMENT NUMBER ISNO = 4040
DEBUG PRINT: "STATEMENT NUMBER 4040" "SAMPLE NUMBER _" PDUMP A6 VALUE
L (4050)
IS THE SAMPLE COUNT < THE INTERVAL START POSITION? NX < PRNDX J (1,1)
Y
K (4040)
N
CHECK TO SEE IF PROCESSING HAS BEEN COMPLETED FOR THIS MEAS ARRAY
7

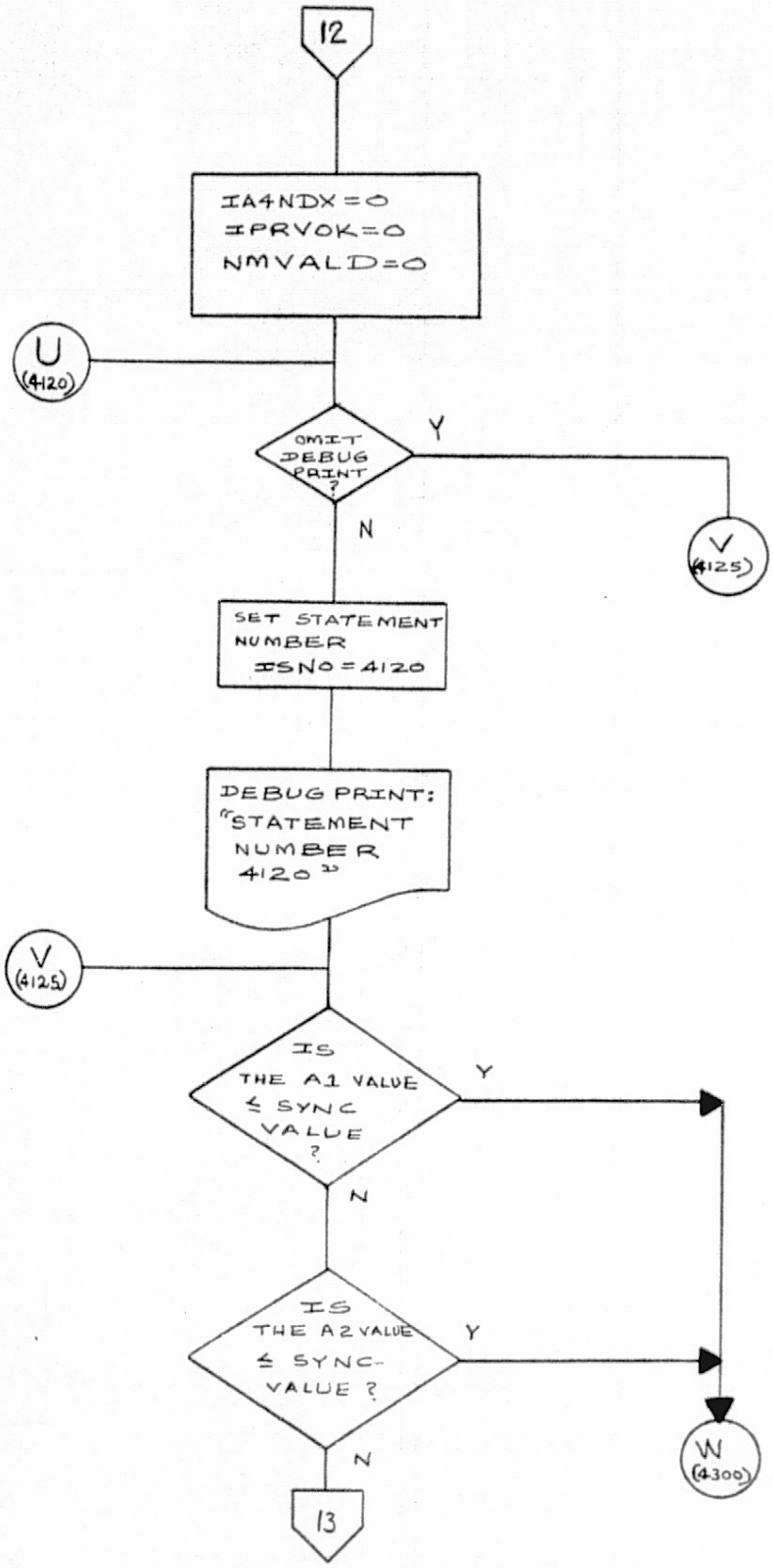

7
IS THE SAMPLE COUNT ≤ TO THE INTERVAL STOP POSITION ?
Y
N
M (4060)
DECREMENT SAMPLE COUNTER
NX = NX - 1
HAS START OF SCAN BEEN FOUND? IBGNSC ≠ 0?
Y
N
N (4200)
OMIT DEBUG PRINT ?
Y
N
O (4059)
SET STATEMENT NUMBER
ISNO = 4040
DEBUG PRINT:
"STATEMENT NUMBER 4040"
"LEAVING RAMPRO"
8

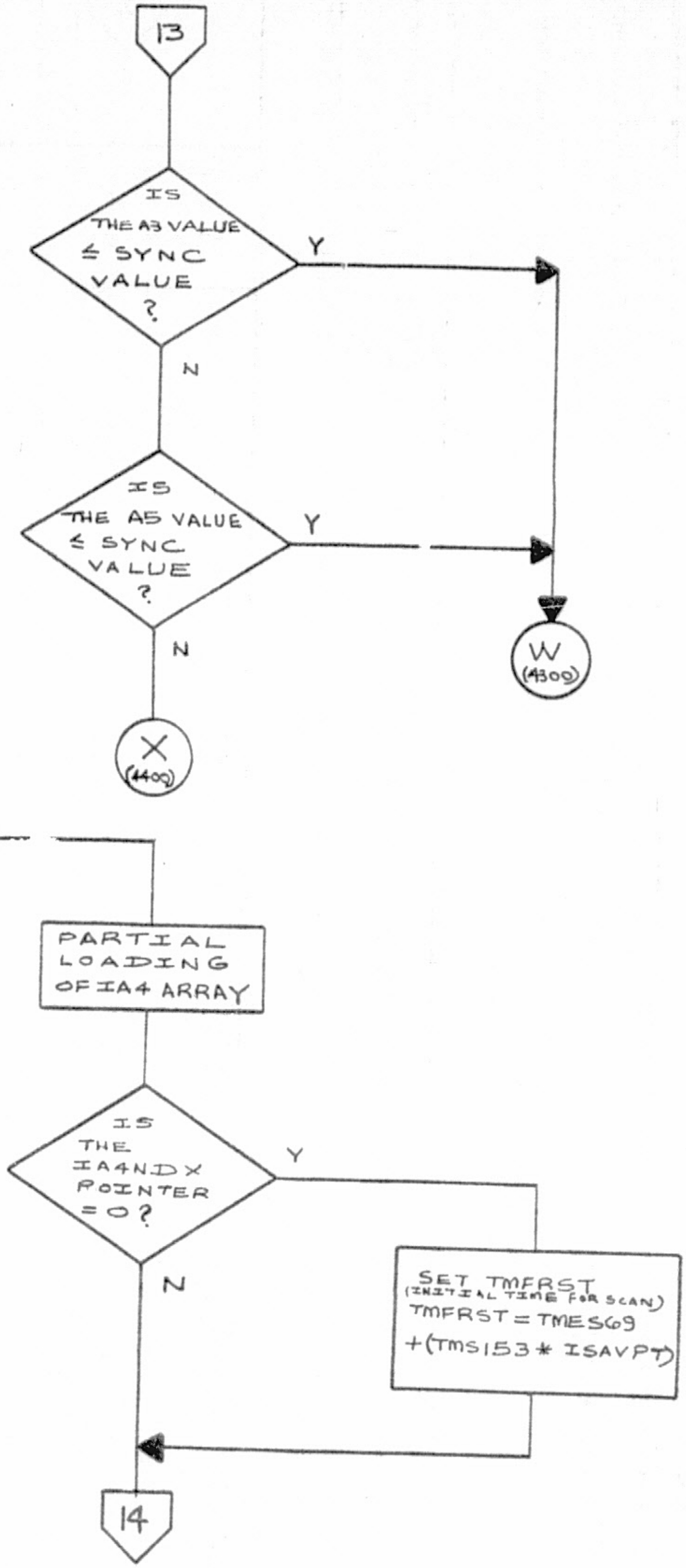

8
O
(4059)
SEG
CALL IN
QAEXEC.GGG
LOAD
MODULE
M
(4060)
IS THE
VALUE OF
A6 GREATER
THAN THE
SYNC VALUE
?
Y
P
(4080)
N
Q
(4061)
IAGSET = 0
ARE
THE NUMBER
OF SAMPLES LESS
THAN OR EQUAL TO
THE MAX ALLOWABLE
FOR A VALID
SCAN
?
Y
H
(4030)
N
SET INDICATORS
FOR USE BY THE
ANPRO LOAD
MODULE
KRET = 2
JPAR = 16
NSAM = NX
LAN = 6
SEG
CALL IN
ANPRO.LLL
LOAD MODULE
FOR THE
OUTPUT OF AN
ANOMALY MESSAGE
9

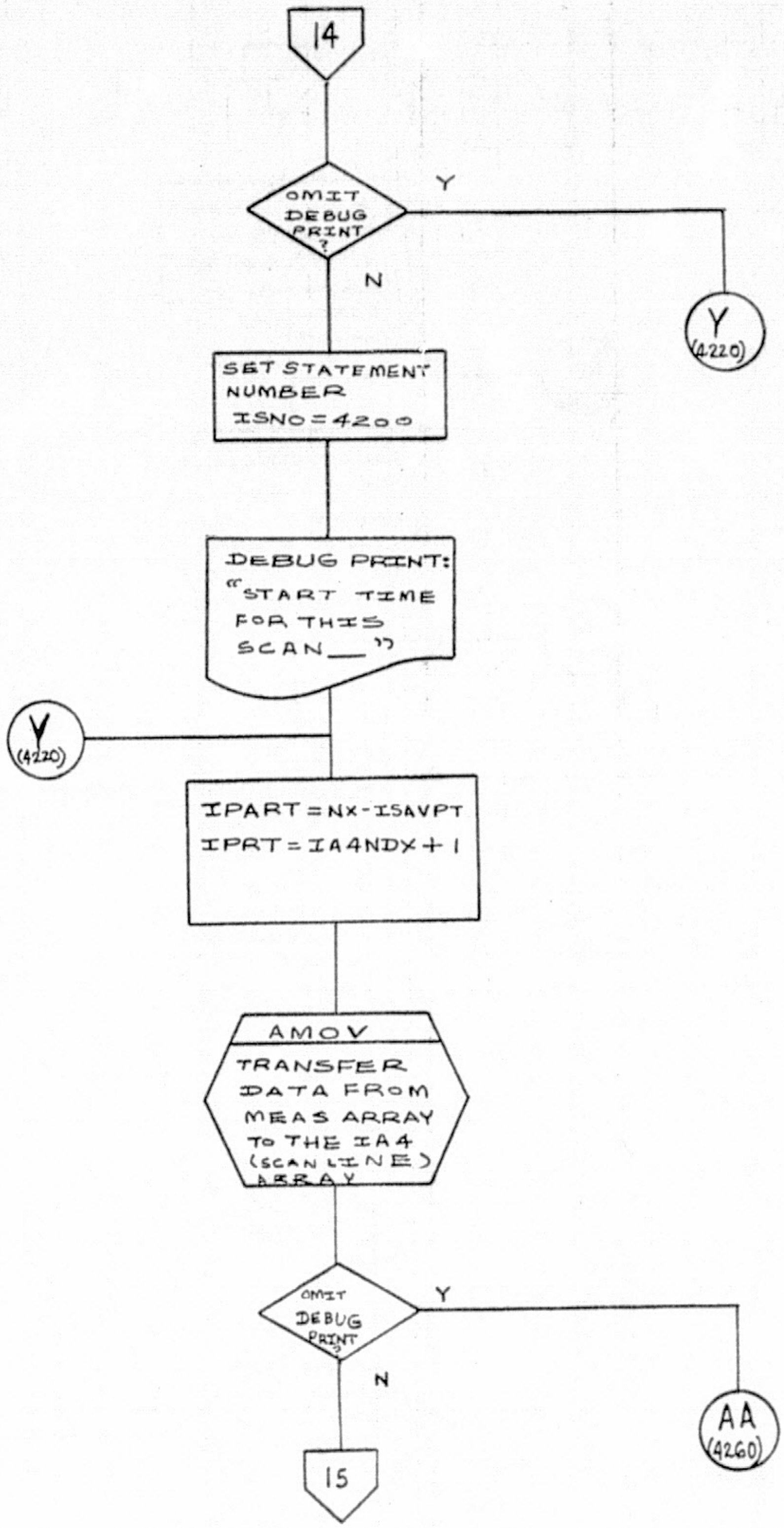

9
ZERO
CLEAR IA4 (SCAN LINE) ARRAY
IBGNSC = 0
IPRVOK = 0
IA4NDX = 0
NMVALD = 0
SUMTM = 0.
USCNTM = 0.
H (4030)
P (4080)
IAGSET = IAGSET + 1
IAGSET = 2?
Y
N
Q (4061)
10

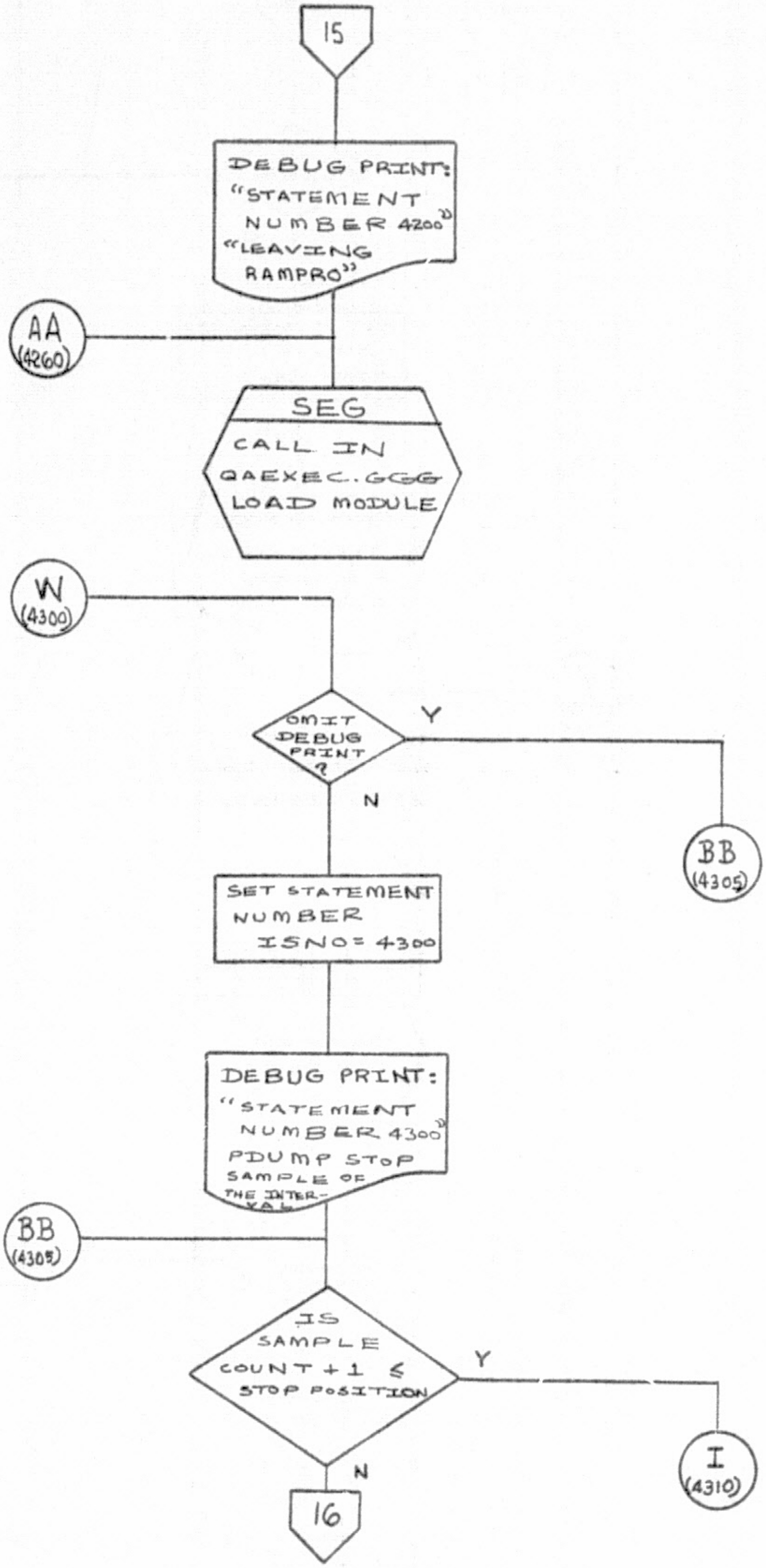

10
ISAVPT - THE LAST POSITION IN THE MEAS ARRAY THAT CONTAINED DATA FOR THE PREVIOUS SCAN
ISAVPT = NX-1
S (4083)
IS SAMPLE NUMBER LESS THAN OR EQUAL TO 1?
Y
T (4085)
N
11

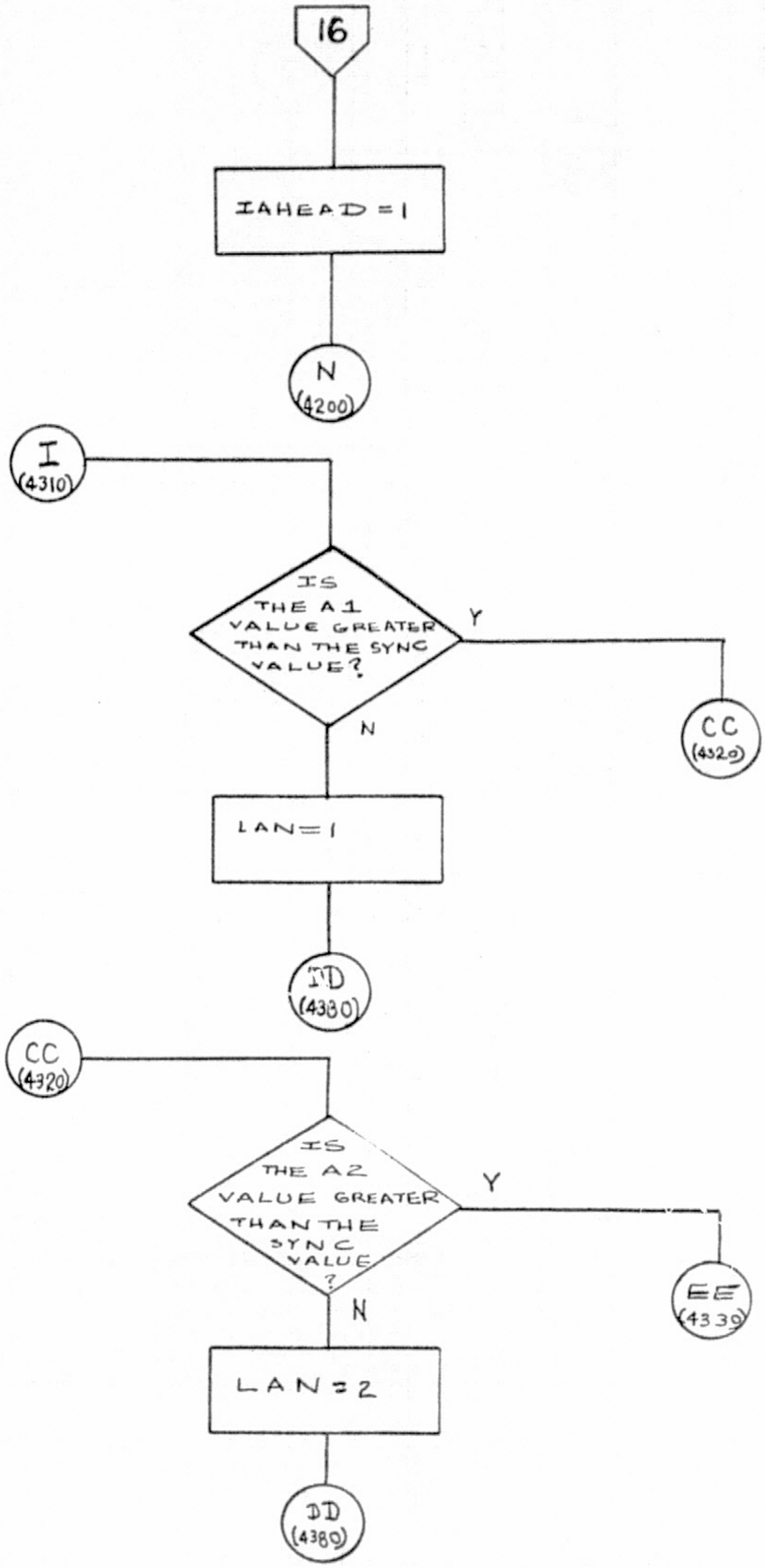

11
IS A4 VALUE GREATER THAN MINIMUM ALLOWABLE VALUE FOR VALID END OF SCAN ?
Y
N
T (4085)
U (4120)
SET INDICATORS FOR USE BY THE ANPRO LOAD MODULE
KRET = 2
JPAR = 18
NSAM = NX
LAN = 2
SEG
CALL IN ANPRO.LLL LOAD MODULE FOR THE OUTPUT OF AN ANOMALY MESSAGE
ZERO
CLEAR THE IA4 (SCAN LINE) ARRAY
12

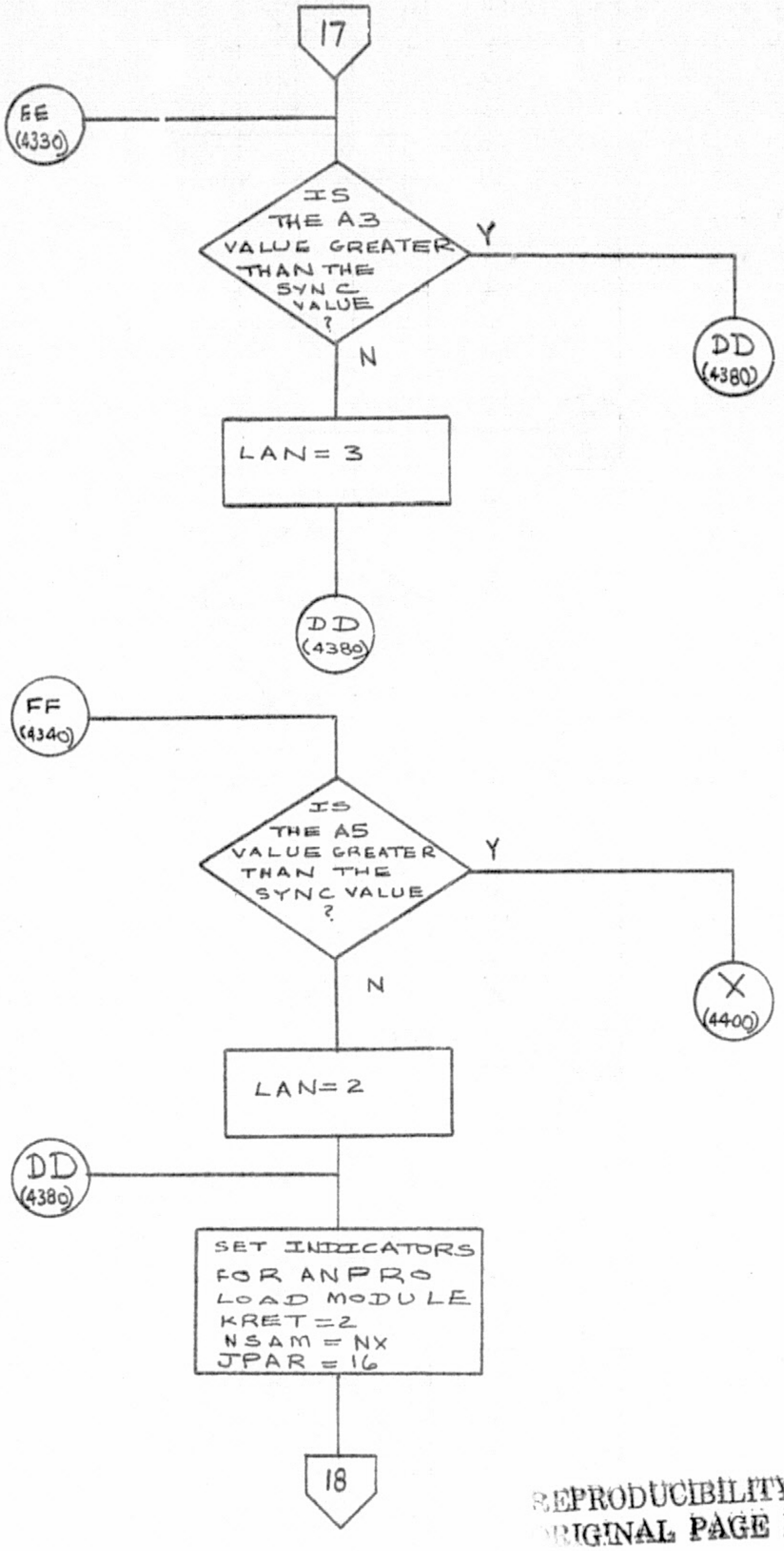

12
IA4NDX=0
IPRVOK=0
NMVALD=0
U
(4120)
OMIT DEBUG PRINT ?
Y
N
V
(4125)
SET STATEMENT NUMBER
ISNO=4120
DEBUG PRINT:
"STATEMENT NUMBER 4120"
V
(4125)
IS THE A1 VALUE ≤ SYNC VALUE ?
Y
N
IS THE A2 VALUE ≤ SYNC-VALUE ?
Y
N
W
(4300)
13

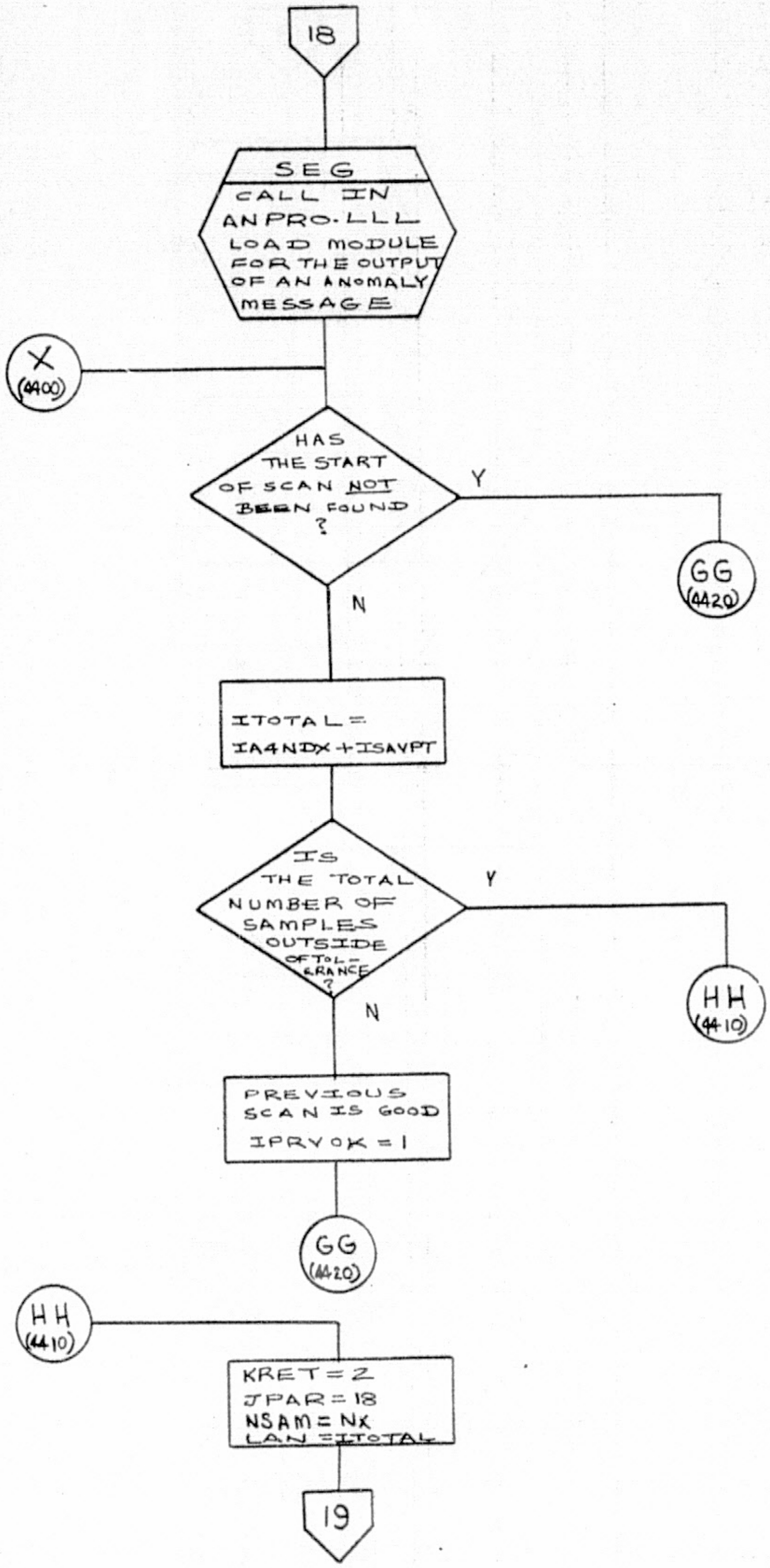

13
IS THE A3 VALUE ≤ SYNC VALUE ?
Y
N
IS THE A5 VALUE ≤ SYNC VALUE ?
Y
N
W (4300)
X (4400)
N (4200)
PARTIAL LOADING OF IA4 ARRAY
IS THE IA4NDX POINTER = 0 ?
Y
N
SET TMFRST (INITIAL TIME FOR SCAN)
TMFRST = TMES69 + (TMS153 * ISAVPT)
14

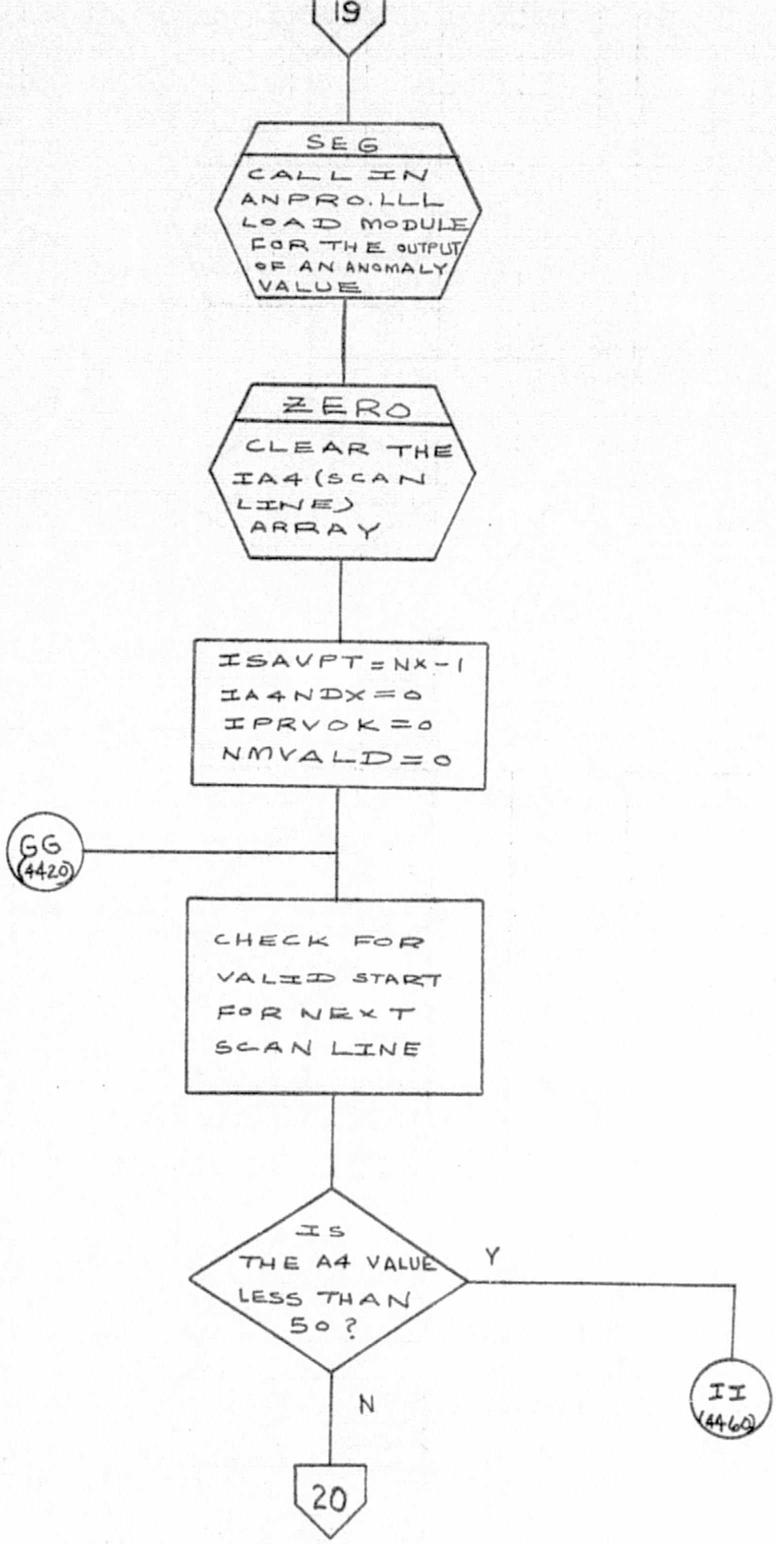

14
OMIT DEBUG PRINT ?
Y
N
Y (4220)
SET STATEMENT NUMBER
ISNO=4200
DEBUG PRINT:
"START TIME FOR THIS SCAN__"
Y (4220)
IPART=NX-ISAVPT
IPRT=IA4NDX+1
AMOV
TRANSFER DATA FROM MEAS ARRAY TO THE IA4 (SCAN LINE) ARRAY
OMIT DEBUG PRINT ?
Y
N
AA (4260)
15

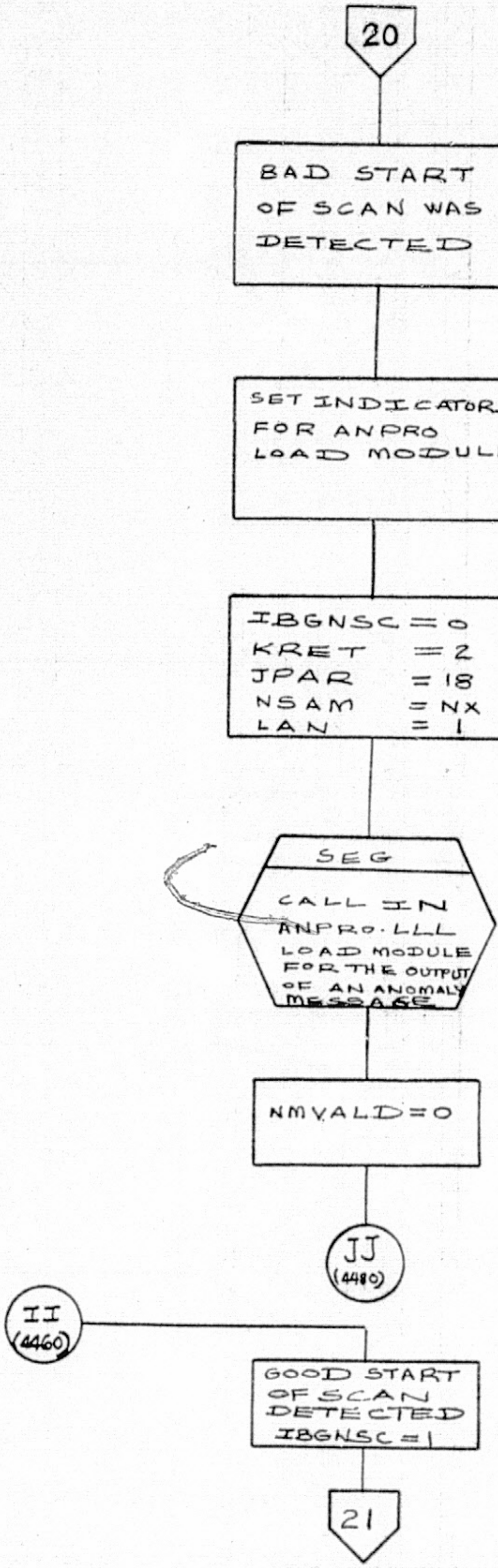

15
DEBUG PRINT: "STATEMENT NUMBER 4200" "LEAVING RAMPRO"
AA (4260)
SEG
CALL IN QAEXEC.GGG LOAD MODULE
W (4300)
OMIT DEBUG PRINT ?
Y
N
BB (4305)
SET STATEMENT NUMBER ISNO= 4300
DEBUG PRINT: "STATEMENT NUMBER 4300" PDUMP STOP SAMPLE OF THE INTER-VAL
BB (4305)
IS SAMPLE COUNT + 1 ≤ STOP POSITION
Y
N
I (4310)
16

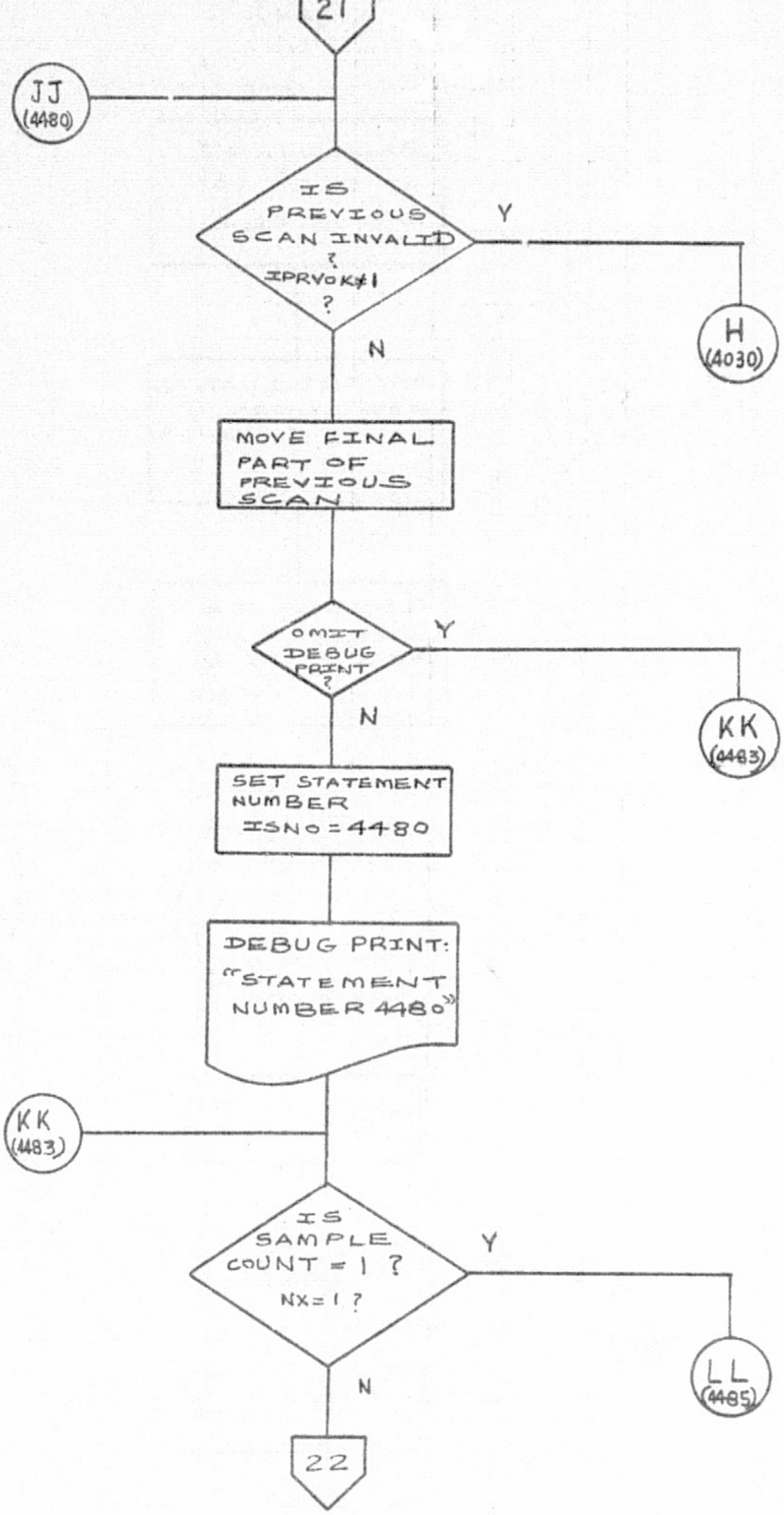

16
IAHEAD = 1
N
(4200)
I
(4310)
IS THE A1 VALUE GREATER THAN THE SYNC VALUE?
Y
N
CC
(4320)
LAN=1
DD
(4380)
CC
(4320)
IS THE A2 VALUE GREATER THAN THE SYNC VALUE?
Y
N
EE
(4330)
LAN=2
DD
(4380)

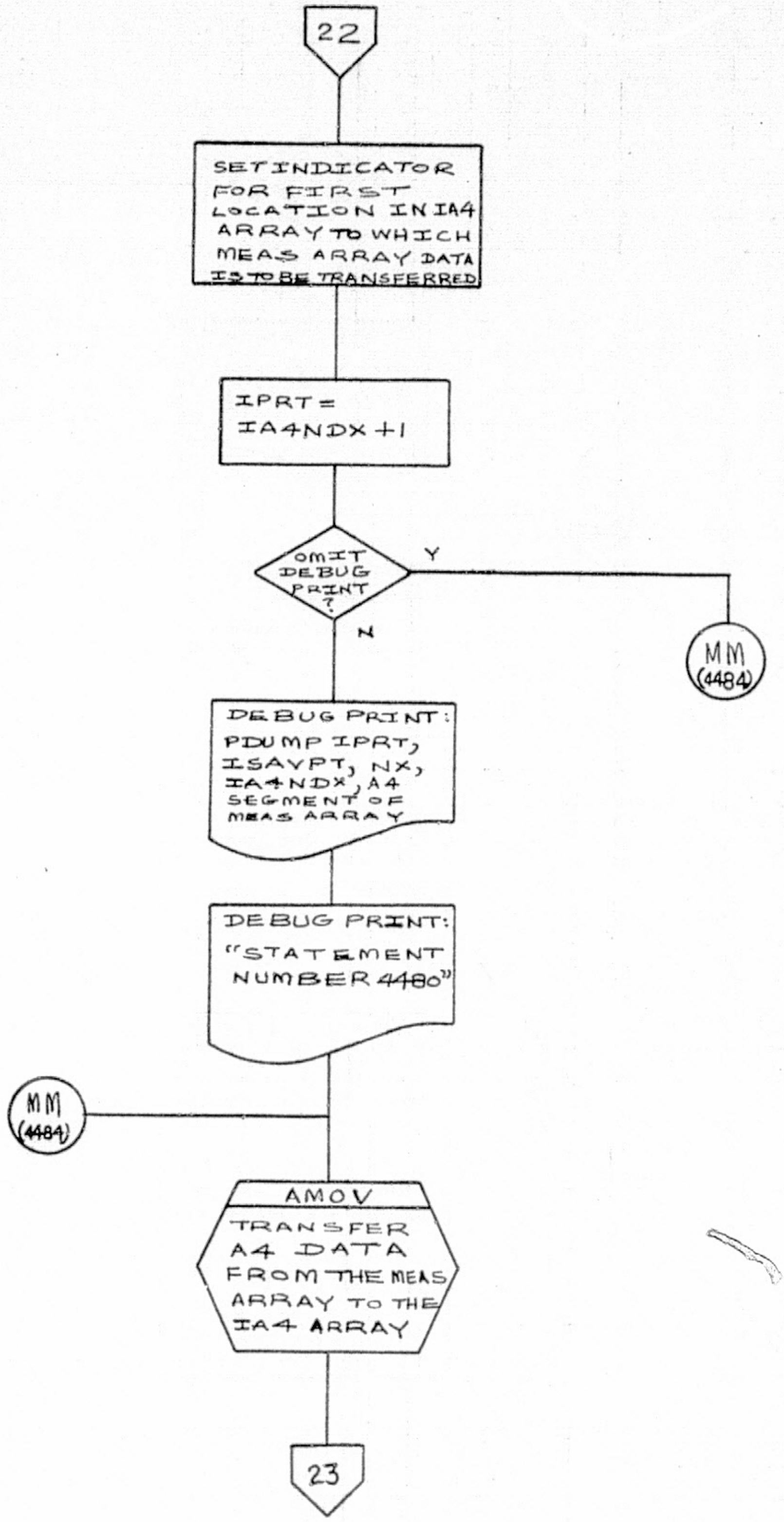

17
EE
(4330)
IS THE A3 VALUE GREATER THAN THE SYNC VALUE ?
Y
N
DD
(4380)
LAN= 3
DD
(4380)
FF
(4340)
IS THE A5 VALUE GREATER THAN THE SYNC VALUE ?
Y
N
X
(4400)
LAN= 2
DD
(4380)
SET INDICATORS
FOR ANPRO
LOAD MODULE
KRET=2
NSAM = NX
JPAR = 16
18

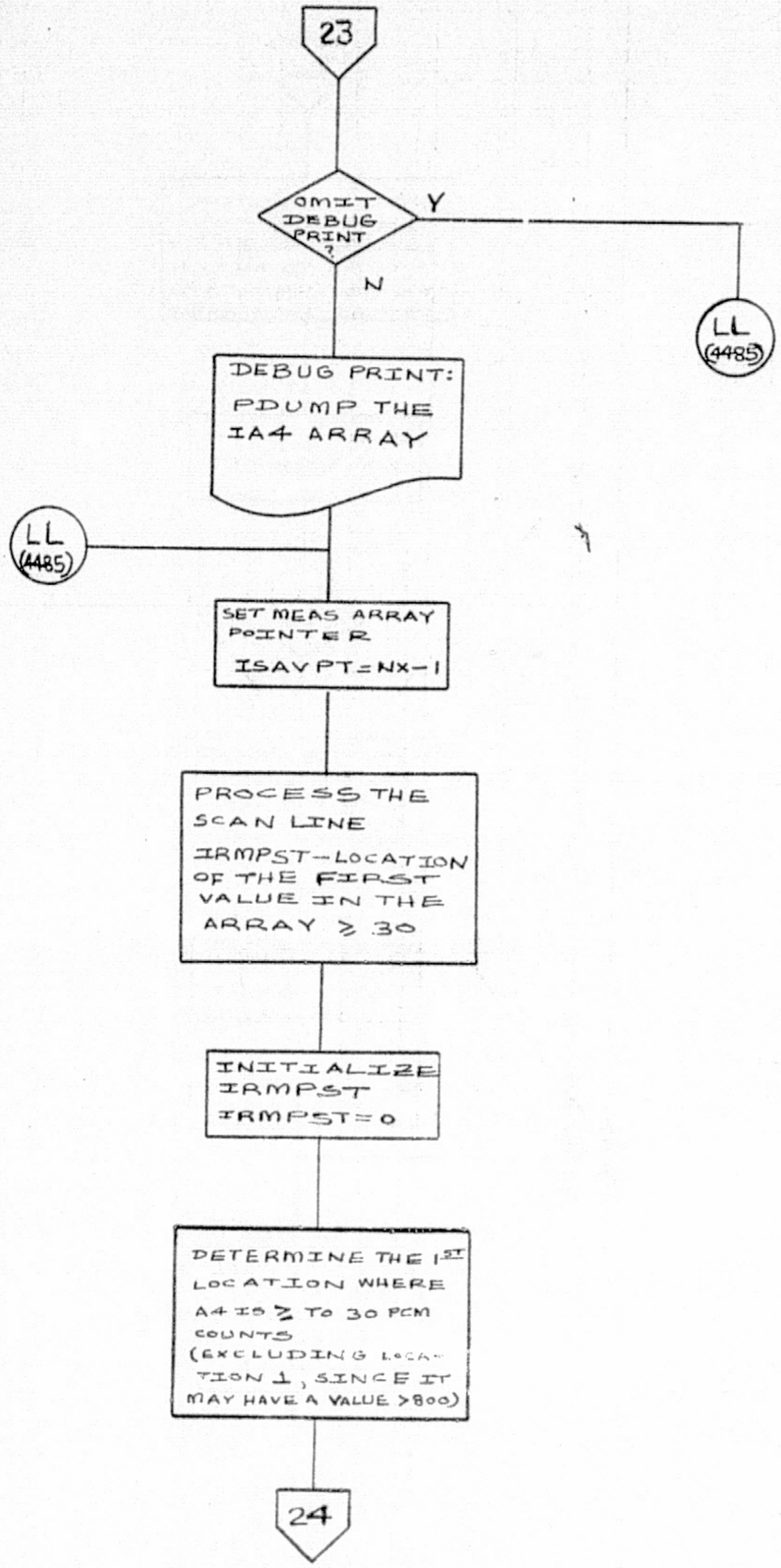

18
SEG
CALL IN ANPRO.LLL LOAD MODULE FOR THE OUTPUT OF AN ANOMALY MESSAGE
X (4400)
HAS THE START OF SCAN NOT BEEN FOUND ?
Y
GG (4420)
N
ITOTAL= IA4NDX+ISAVPT
IS THE TOTAL NUMBER OF SAMPLES OUTSIDE OF TOLERANCE ?
Y
HH (4410)
N
PREVIOUS SCAN IS GOOD
IPRVOK=1
GG (4420)
HH (4410)
KRET=2
JPAR=18
NSAM=NX
LAN=ITOTAL
19

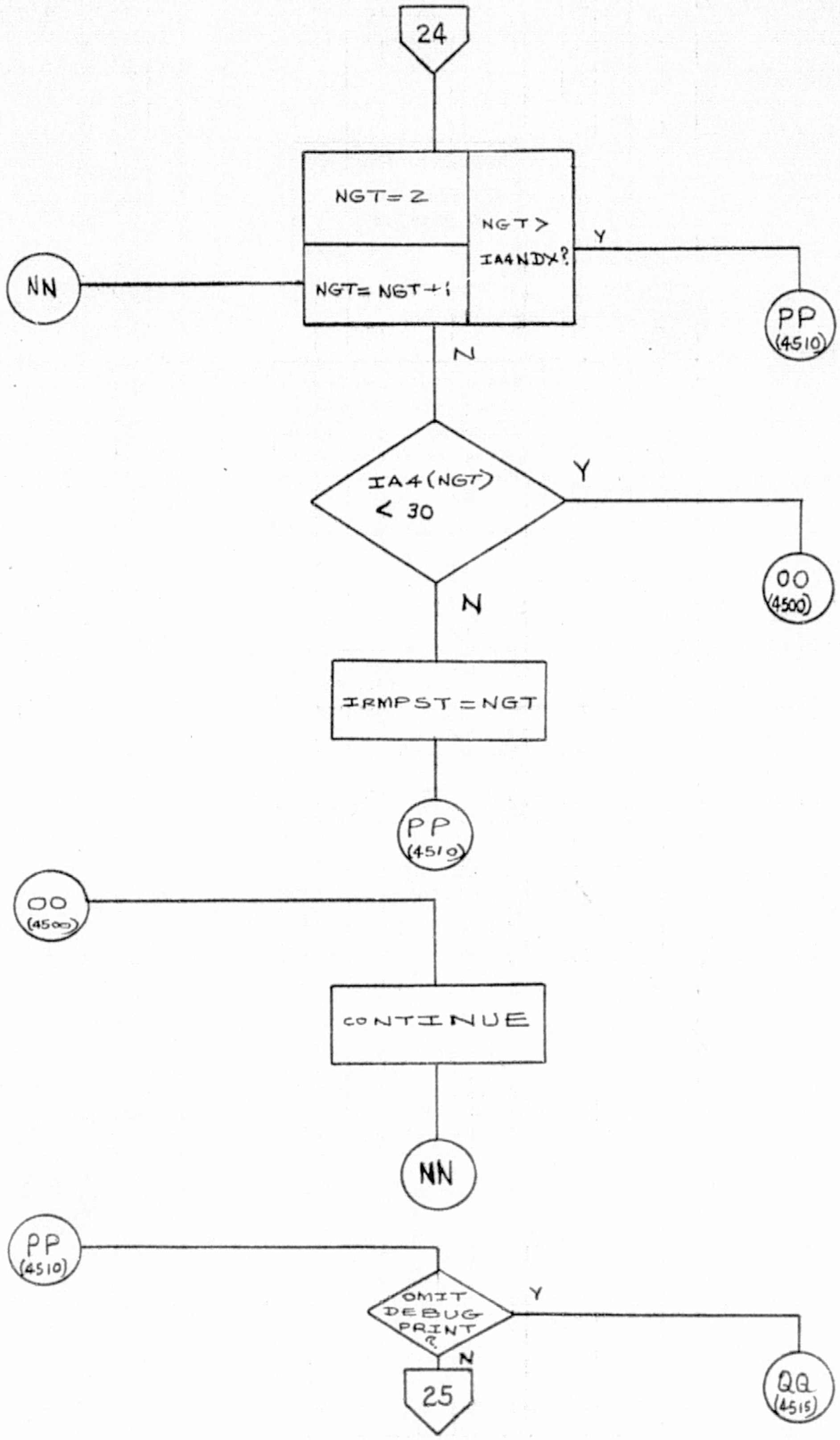

19
SEG
CALL IN ANPRO.LLL LOAD MODULE FOR THE OUTPUT OF AN ANOMALY VALUE
ZERO
CLEAR THE IA4 (SCAN LINE) ARRAY
ISAVPT = NX - 1
IA4NDX = 0
IPRVOK = 0
NMVALD = 0
GG
(4420)
CHECK FOR VALID START FOR NEXT SCAN LINE
IS THE A4 VALUE LESS THAN 50 ?
Y
N
II
(4460)
20

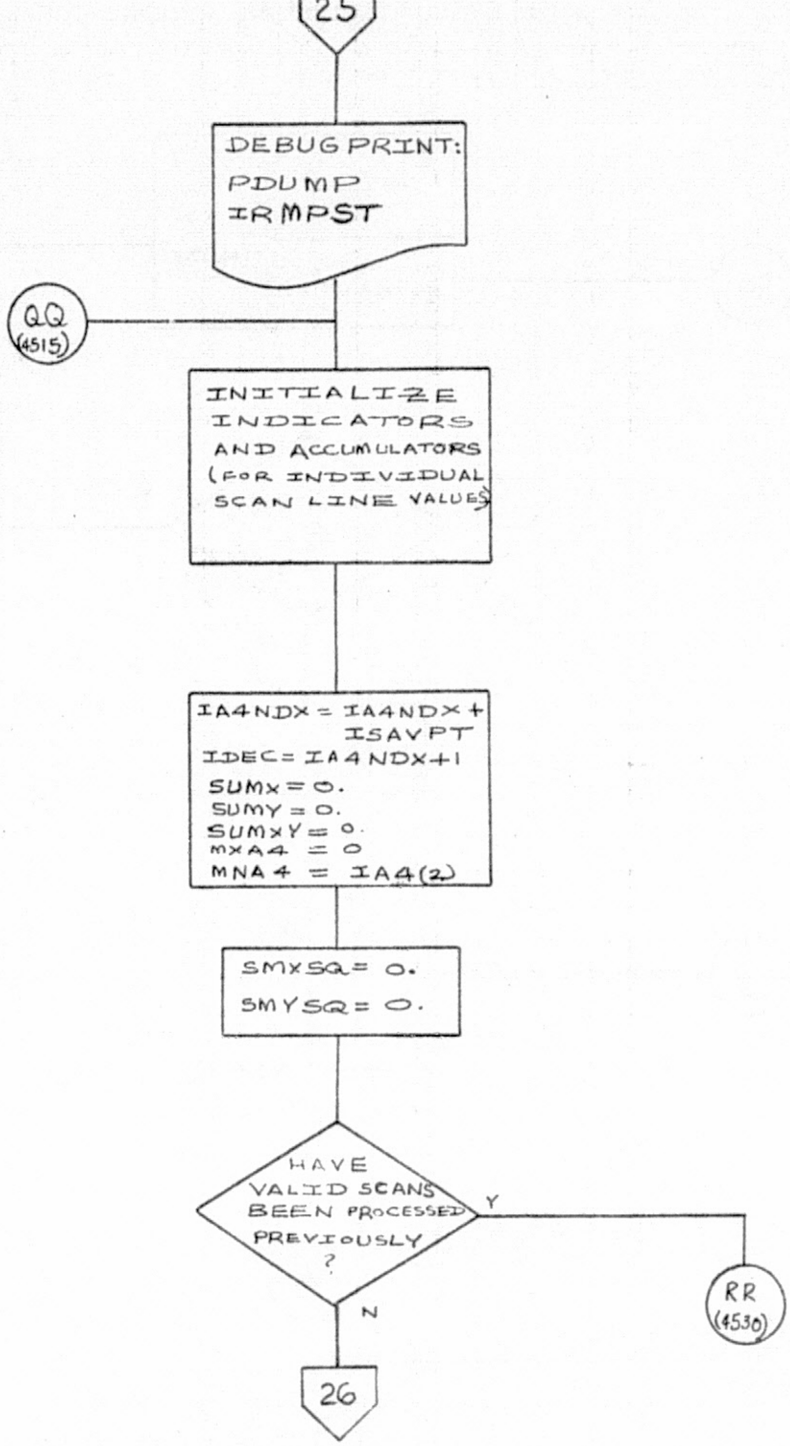

20
BAD START OF SCAN WAS DETECTED
SET INDICATORS FOR ANPRO LOAD MODULE
IBGNSC = 0
KRET = 2
JPAR = 18
NSAM = NX
LAN = 1
SEG
CALL IN ANPRO.LLL LOAD MODULE FOR THE OUTPUT OF AN ANOMALY MESSAGE
NMVALD=0
JJ
(4480)
II
(4460)
GOOD START OF SCAN DETECTED
IBGNSC=1
21

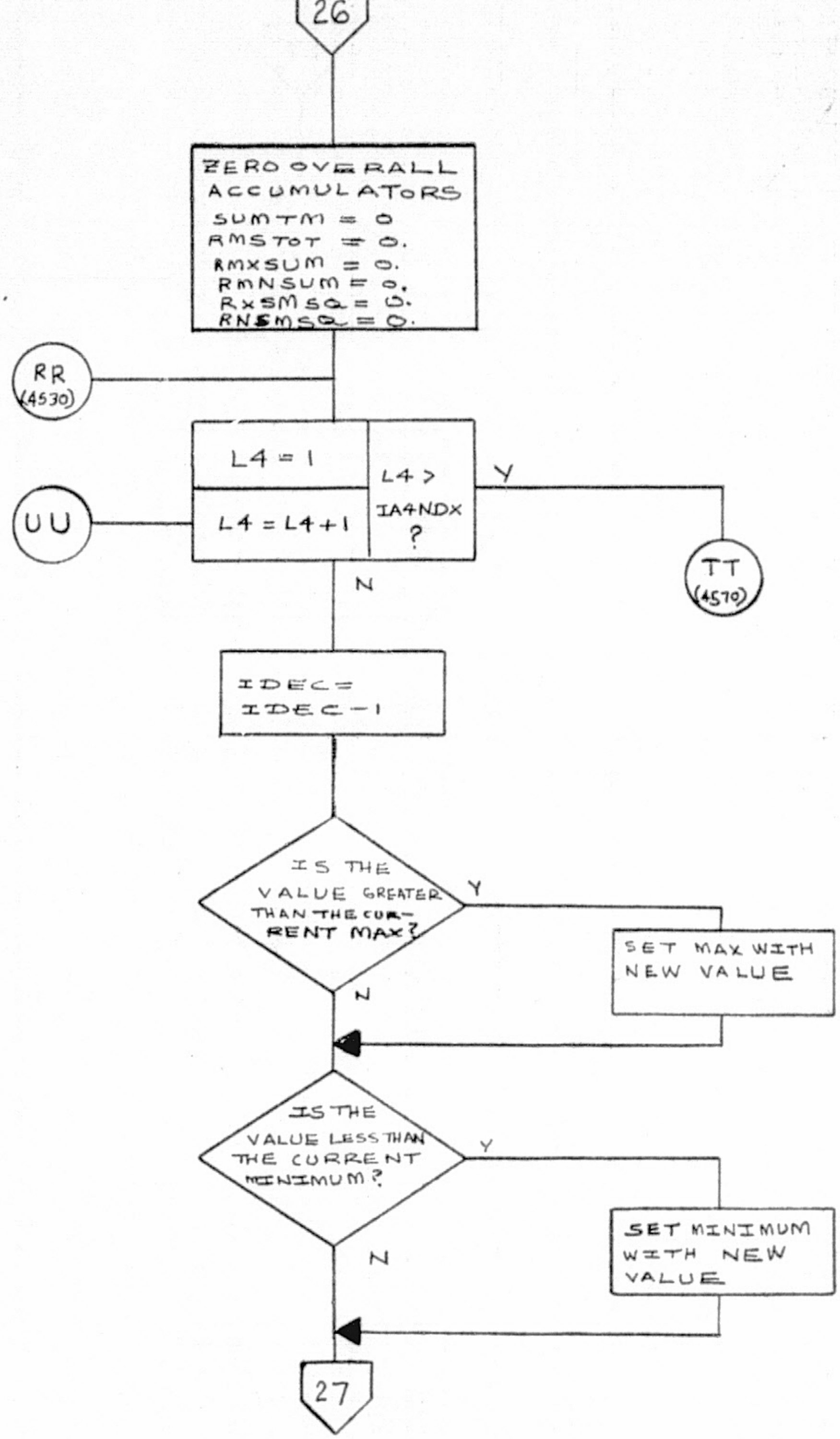

21
JJ (4480)
IS PREVIOUS SCAN INVALID ? IPRVOK≠1 ?
Y
H (4030)
N
MOVE FINAL PART OF PREVIOUS SCAN
OMIT DEBUG PRINT ?
Y
KK (4483)
N
SET STATEMENT NUMBER ISNO=4480
DEBUG PRINT: "STATEMENT NUMBER 4480"
KK (4483)
IS SAMPLE COUNT = 1 ? NX=1 ?
Y
LL (4485)
N
22

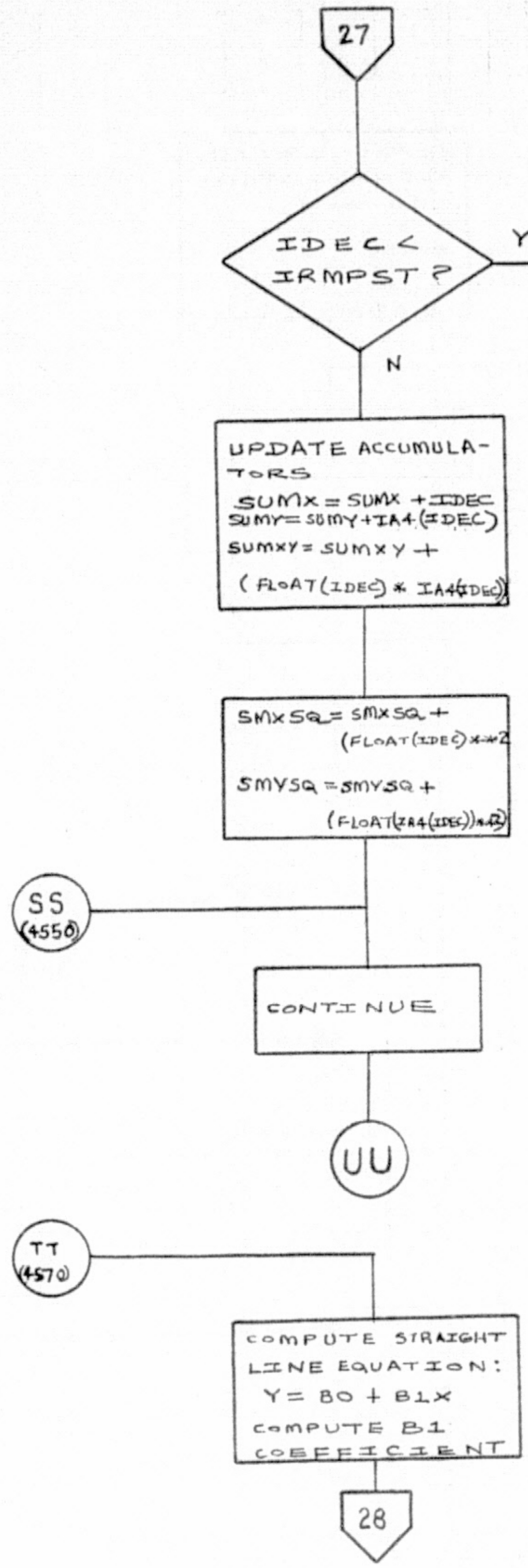

22
SET INDICATOR FOR FIRST LOCATION IN IA4 ARRAY TO WHICH MEAS ARRAY DATA IS TO BE TRANSFERRED
IPRT = IA4NDX + 1
OMIT DEBUG PRINT ?
Y
N
MM (4484)
DEBUG PRINT: PDUMP IPRT, ISAVPT, NX, IA4NDX, A4 SEGMENT OF MEAS ARRAY
DEBUG PRINT: "STATEMENT NUMBER 4480"
MM (4484)
AMOV
TRANSFER A4 DATA FROM THE MEAS ARRAY TO THE IA4 ARRAY
23

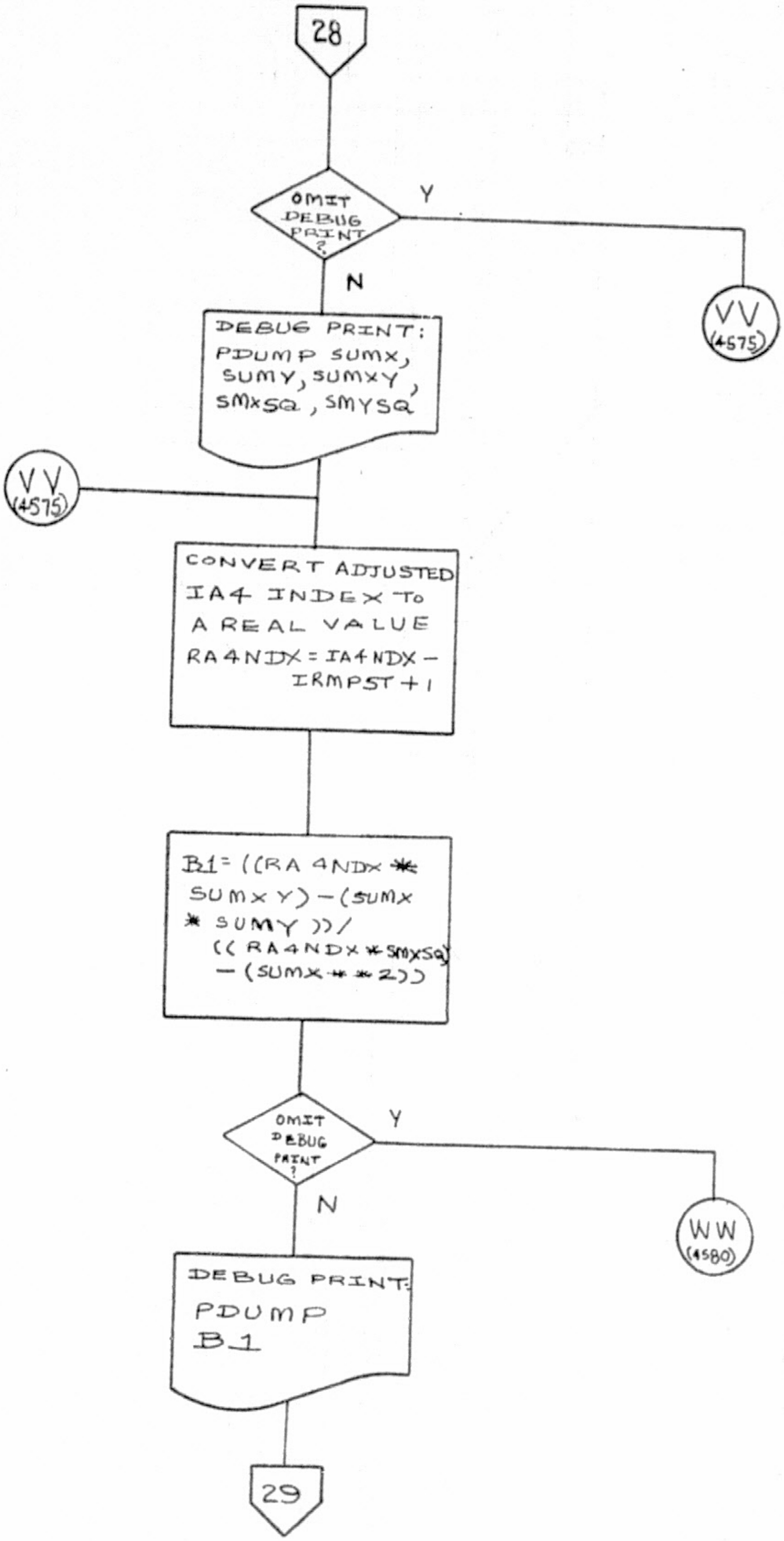

23
OMIT DEBUG PRINT ?
Y
N
LL (4485)
DEBUG PRINT: PDUMP THE IA4 ARRAY
LL (4485)
SET MEAS ARRAY POINTER ISAVPT=NX-1
PROCESS THE SCAN LINE IRMPST--LOCATION OF THE FIRST VALUE IN THE ARRAY ≥ 30
INITIALIZE IRMPST IRMPST=0
DETERMINE THE 1ST LOCATION WHERE A4 IS ≥ TO 30 PCM COUNTS (EXCLUDING LOCATION 1, SINCE IT MAY HAVE A VALUE >800)
24

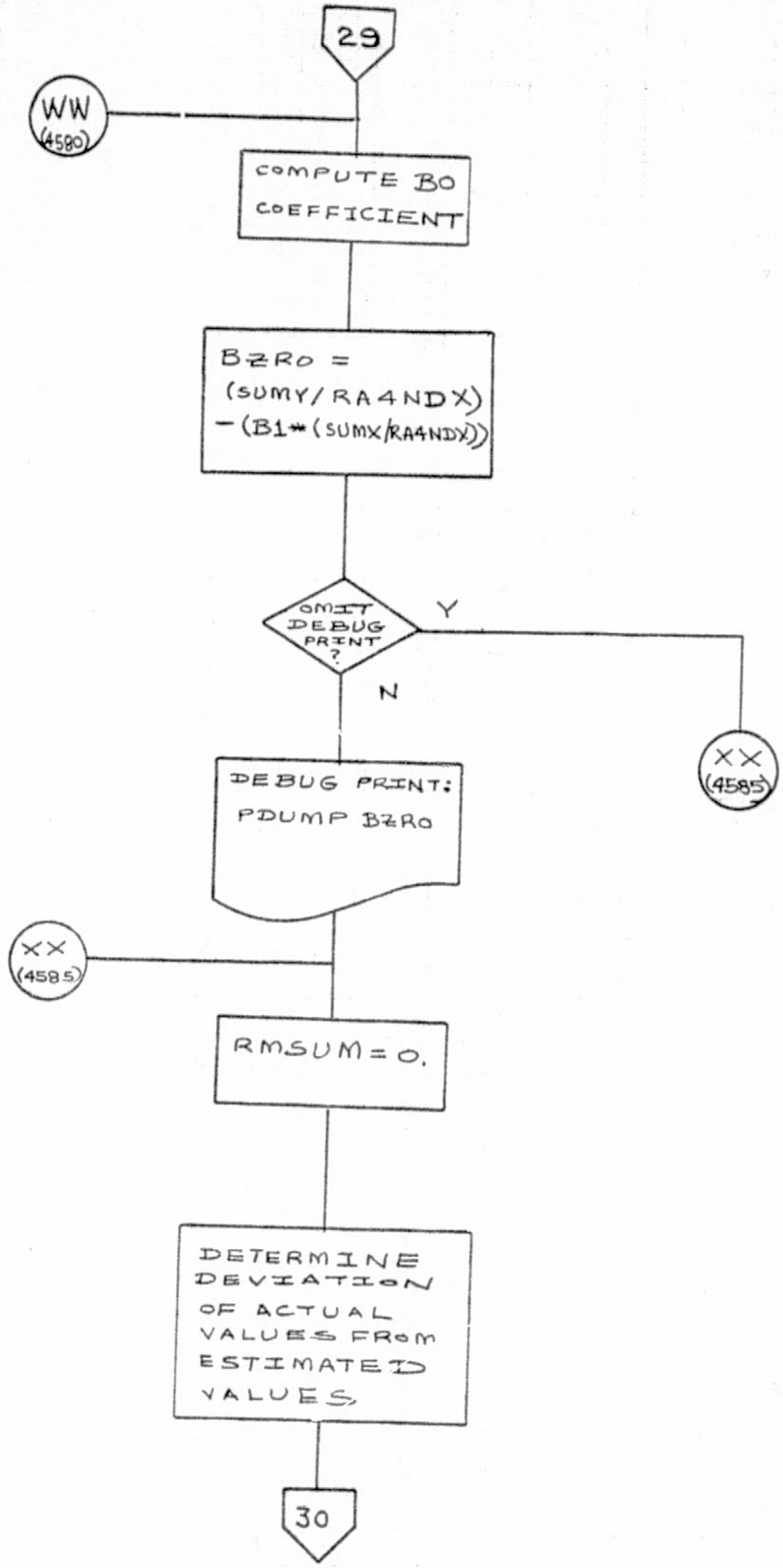

24
NGT= 2
NGT= NGT+1
NGT > IA4NDX?
Y
N
NN
PP (4510)
IA4(NGT) < 30
Y
N
OO (4500)
IRMPST = NGT
PP (4510)
OO (4500)
CONTINUE
NN
PP (4510)
OMIT DEBUG PRINT ?
Y
N
25
QQ (4515)

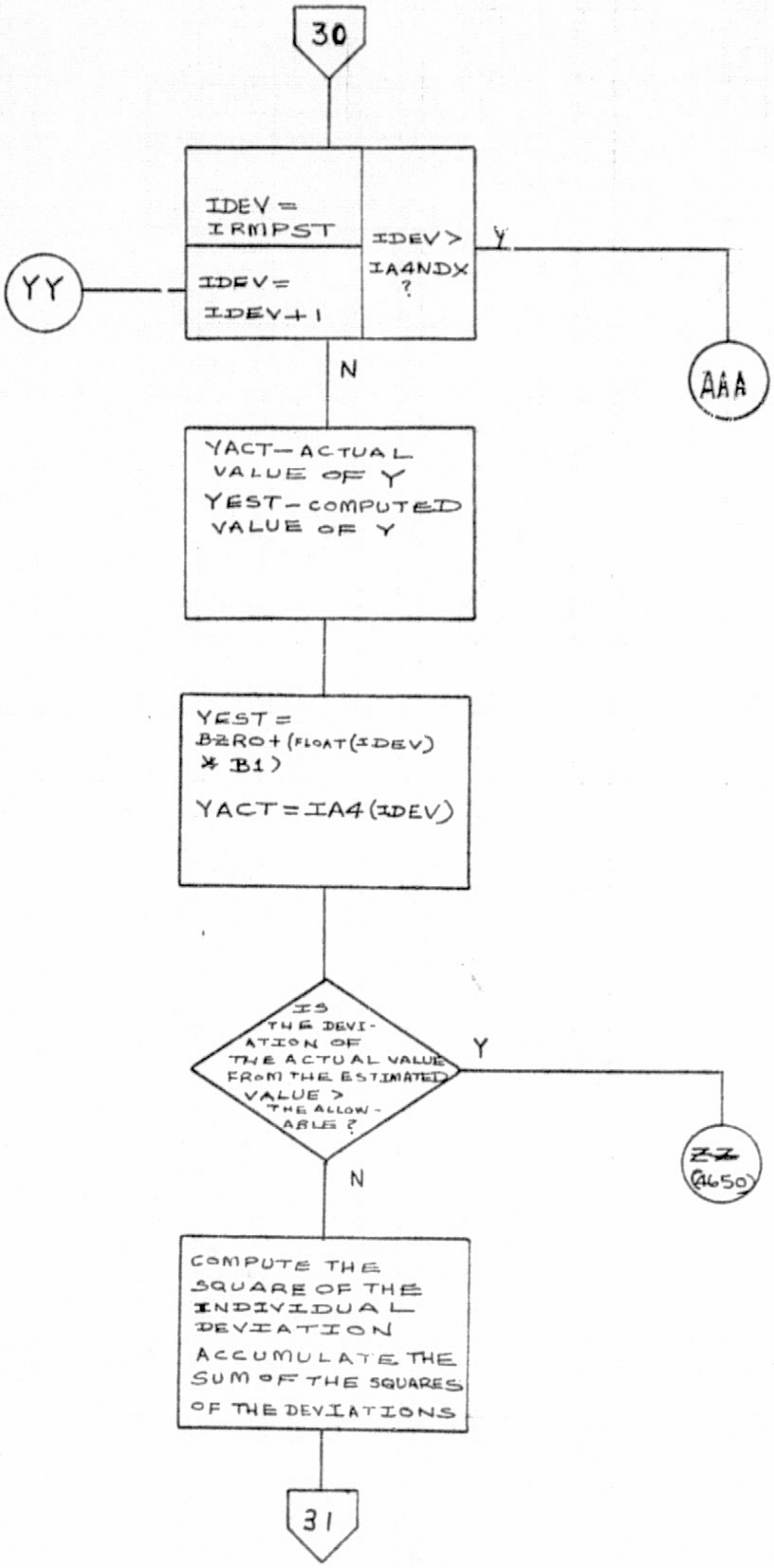

25
DEBUG PRINT:
PDUMP
IRMPST
QQ
(4515)
INITIALIZE
INDICATORS
AND ACCUMULATORS
(FOR INDIVIDUAL
SCAN LINE VALUES)
IA4NDX = IA4NDX +
ISAVPT
IDEC = IA4NDX+1
SUMX = 0.
SUMY = 0.
SUMXY = 0.
MXA4 = 0
MNA4 = IA4(2)
SMXSQ = 0.
SMYSQ = 0.
HAVE
VALID SCANS
BEEN PROCESSED
PREVIOUSLY
?
Y
N
RR
(4530)
26

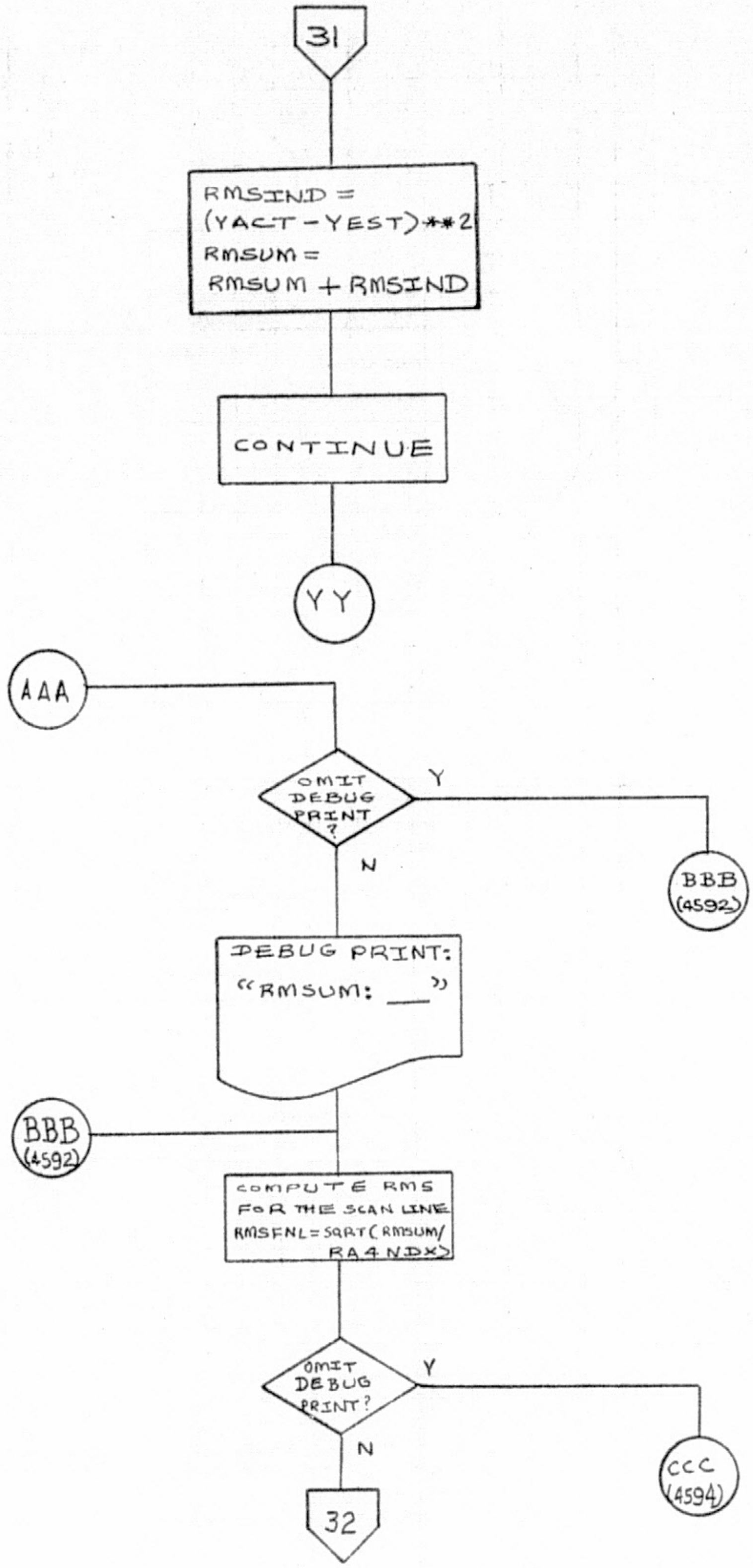

26
ZERO OVERALL ACCUMULATORS
SUMTM = 0
RMSTOT = 0.
RMXSUM = 0.
RMNSUM = 0.
RXSMSQ = 0.
RNSMSQ = 0.
RR (4530)
L4 = 1
L4 = L4+1
L4 > IA4NDX ?
Y
UU
TT (4570)
N
IDEC = IDEC - 1
IS THE VALUE GREATER THAN THE CURRENT MAX?
Y
SET MAX WITH NEW VALUE
N
IS THE VALUE LESS THAN THE CURRENT MINIMUM?
Y
SET MINIMUM WITH NEW VALUE
N
27

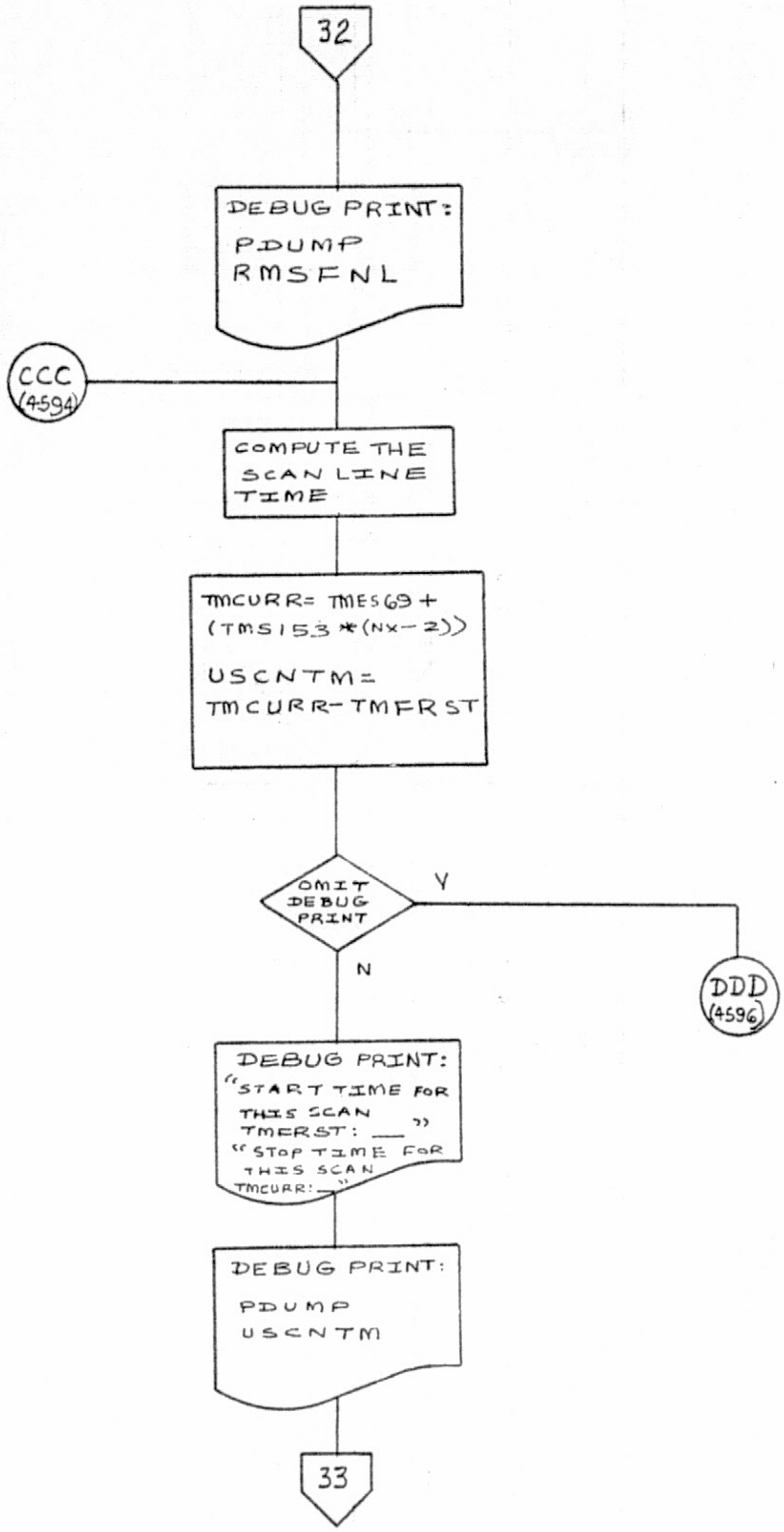

27
IDEC < IRMPST?
Y
N
SS (4550)
UPDATE ACCUMULATORS
SUMX = SUMX + IDEC
SUMY = SUMY + IA4(IDEC)
SUMXY = SUMXY + (FLOAT(IDEC) * IA4(IDEC))
SMXSQ = SMXSQ + (FLOAT(IDEC))**2
SMYSQ = SMYSQ + (FLOAT(IA4(IDEC)))**2
SS (4550)
CONTINUE
UU
TT (4570)
COMPUTE STRAIGHT LINE EQUATION:
Y = B0 + B1X
COMPUTE B1 COEFFICIENT
28

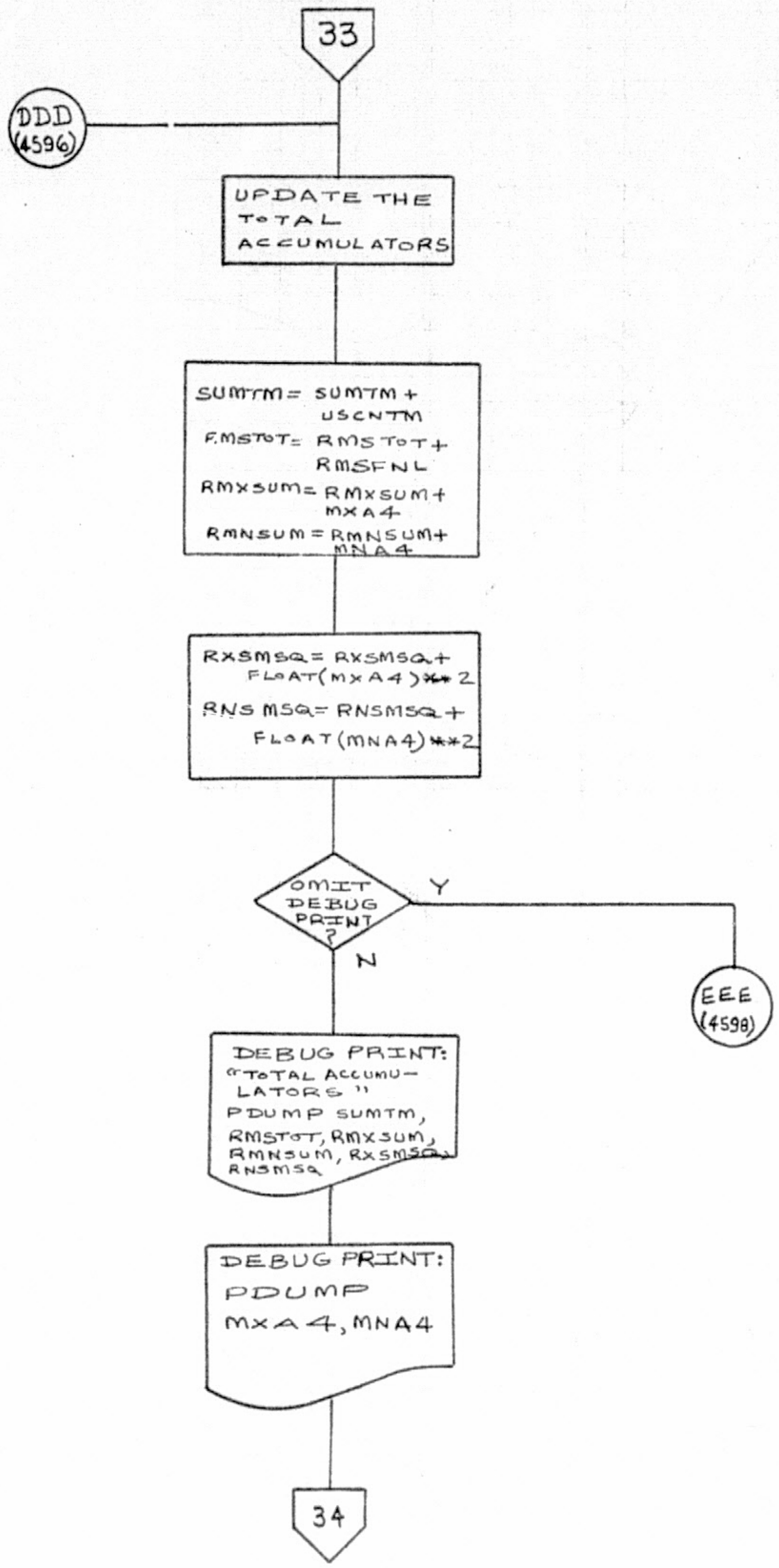

28
OMIT DEBUG PRINT ?
Y
N
VV (4575)
DEBUG PRINT: PDUMP SUMX, SUMY, SUMXY, SMXSQ, SMYSQ
VV (4575)
CONVERT ADJUSTED IA4 INDEX TO A REAL VALUE RA4NDX = IA4NDX - IRMPST + 1
B1 = ((RA4NDX * SUMXY) - (SUMX * SUMY))/ ((RA4NDX * SMXSQ) - (SUMX ** 2))
OMIT DEBUG PRINT ?
Y
N
WW (4580)
DEBUG PRINT: PDUMP B1
29

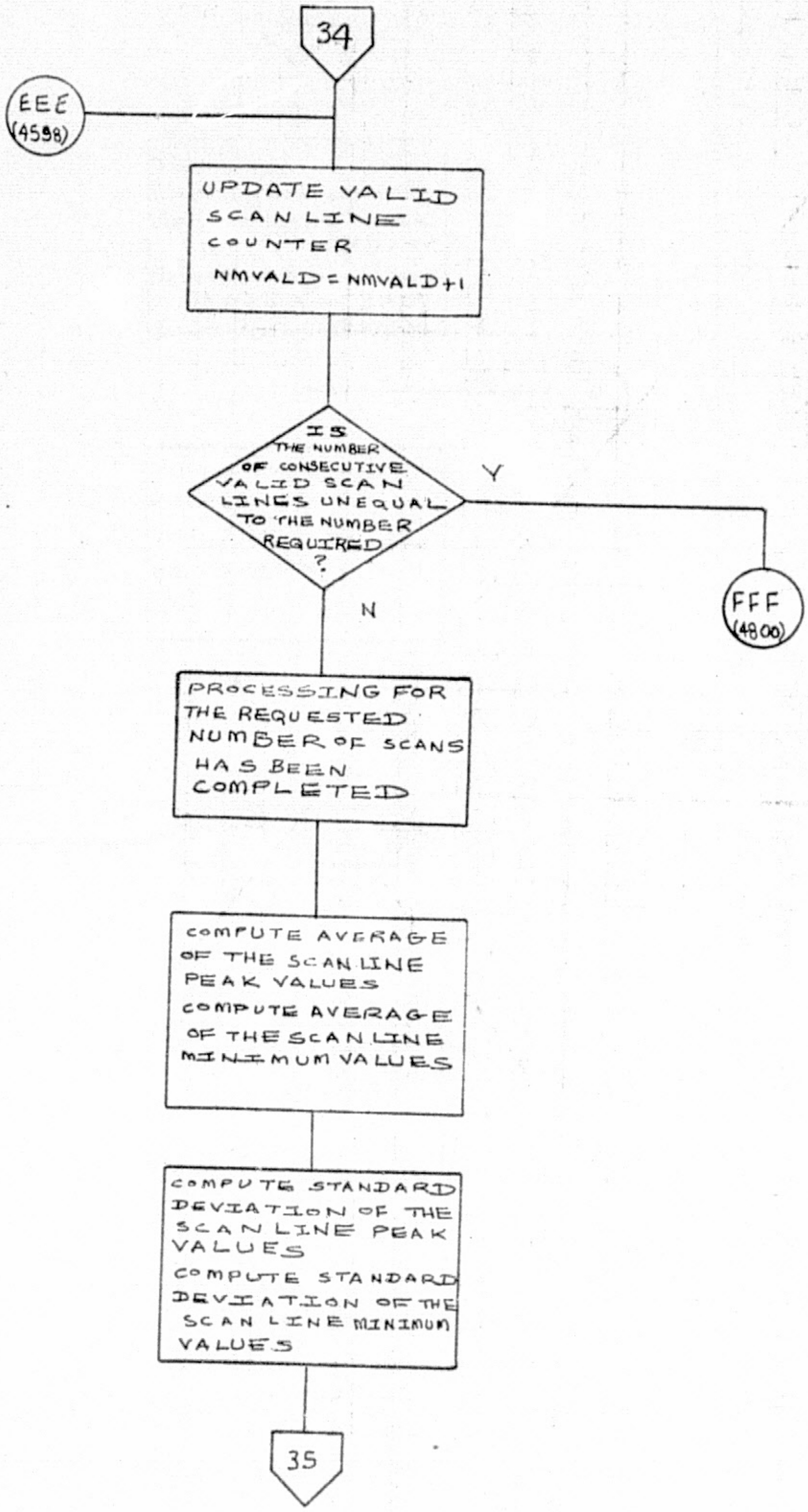

29
WW
(4580)
COMPUTE BO
COEFFICIENT
BZRO =
(SUMY/RA4NDX)
-(B1*(SUMX/RA4NDX))
OMIT
DEBUG
PRINT
?
Y
N
XX
(4585)
DEBUG PRINT:
PDUMP BZRO
XX
(4585)
RMSUM=0.
DETERMINE
DEVIATION
OF ACTUAL
VALUES FROM
ESTIMATED
VALUES.
30

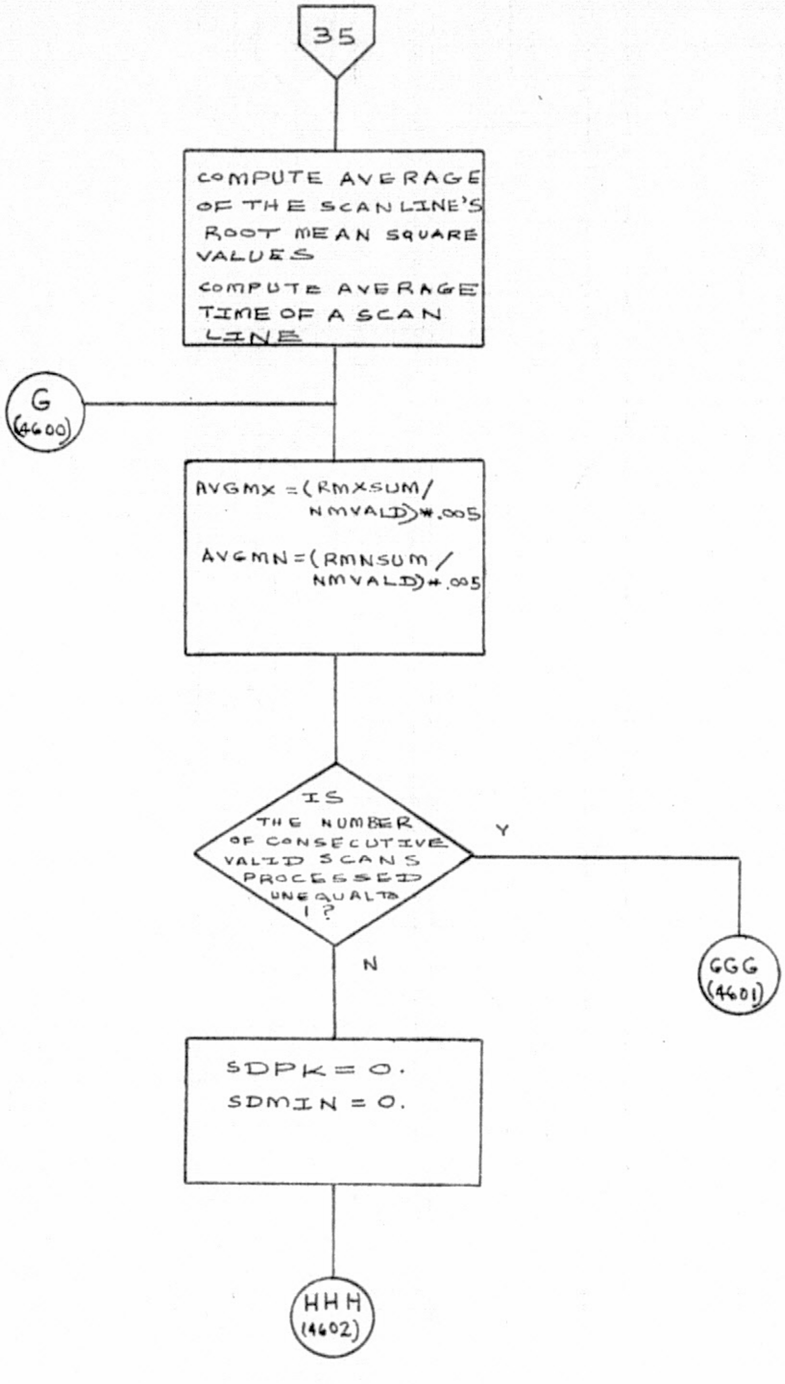

30
IDEV = IRMPST
IDEV = IDEV+1
IDEV > IA4NDX ?
Y
N
YY
AAA
YACT—ACTUAL VALUE OF Y
YEST—COMPUTED VALUE OF Y
YEST = BZRO+(FLOAT(IDEV) * B1)
YACT=IA4(IDEV)
IS THE DEVIATION OF THE ACTUAL VALUE FROM THE ESTIMATED VALUE > THE ALLOWABLE ?
Y
N
ZZ (4650)
COMPUTE THE SQUARE OF THE INDIVIDUAL DEVIATION
ACCUMULATE THE SUM OF THE SQUARES OF THE DEVIATIONS
31

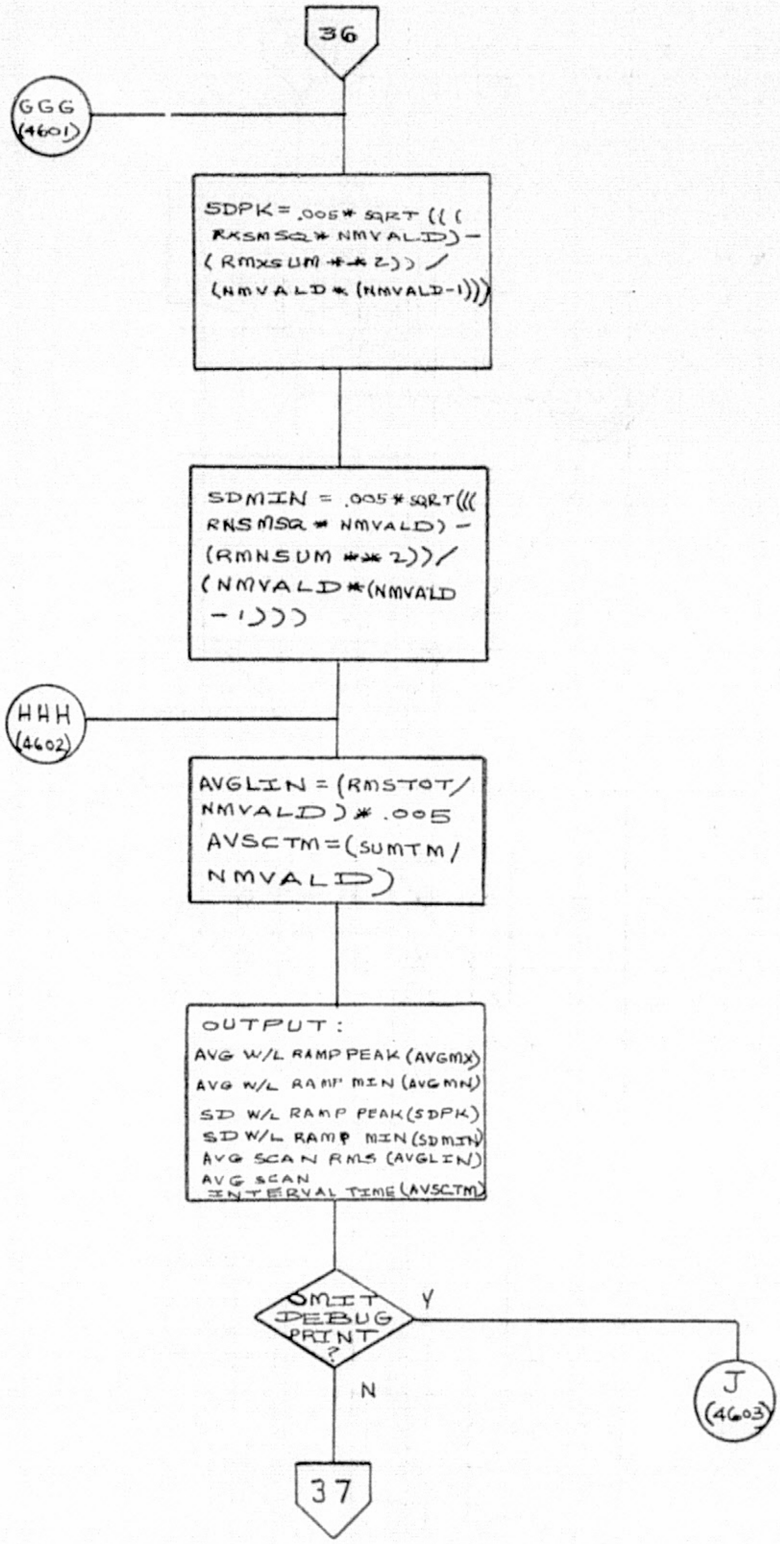

31
RMSIND =
(YACT - YEST)**2
RMSUM =
RMSUM + RMSIND
CONTINUE
YY
AAA
OMIT DEBUG PRINT ?
Y
N
BBB (4592)
DEBUG PRINT:
"RMSUM: ___"
BBB (4592)
COMPUTE RMS FOR THE SCAN LINE
RMSFNL = SQRT(RMSUM/RA4NDX)
OMIT DEBUG PRINT?
Y
N
CCC (4594)
32

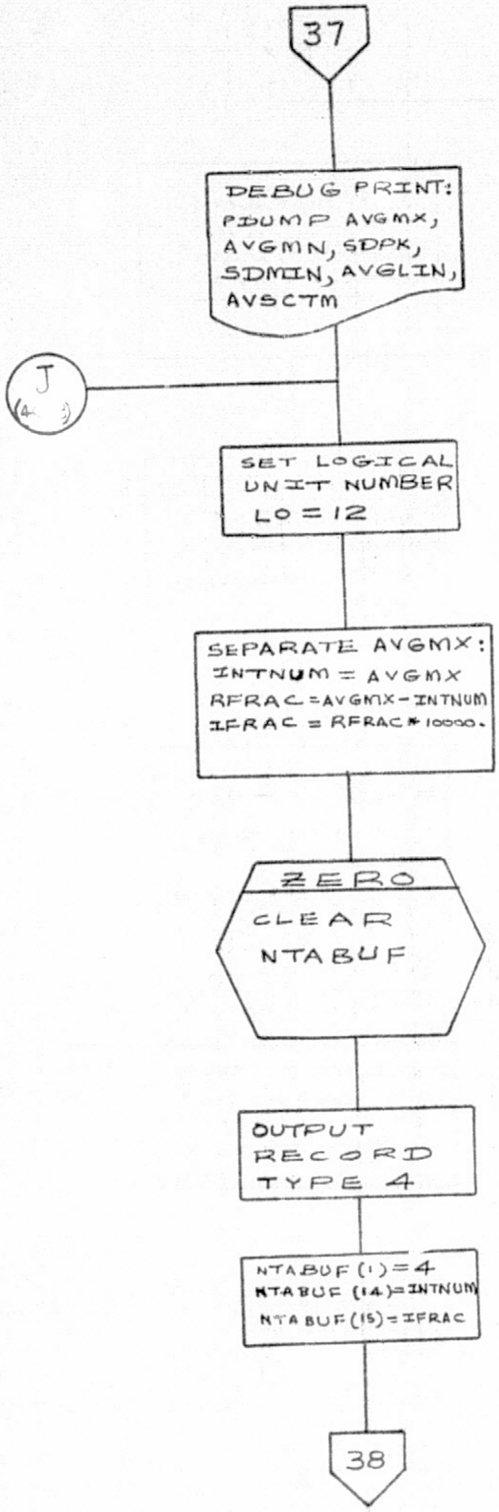

32
DEBUG PRINT:
PDUMP
RMSFNL
CCC
(4594)
COMPUTE THE SCAN LINE TIME
TMCURR= TMES69 + (TMS153 *(NX-2))
USCNTM= TMCURR-TMFRST
OMIT DEBUG PRINT
Y
N
DDD
(4596)
DEBUG PRINT:
"START TIME FOR THIS SCAN TMFRST: __"
"STOP TIME FOR THIS SCAN TMCURR: __"
DEBUG PRINT:
PDUMP
USCNTM
33

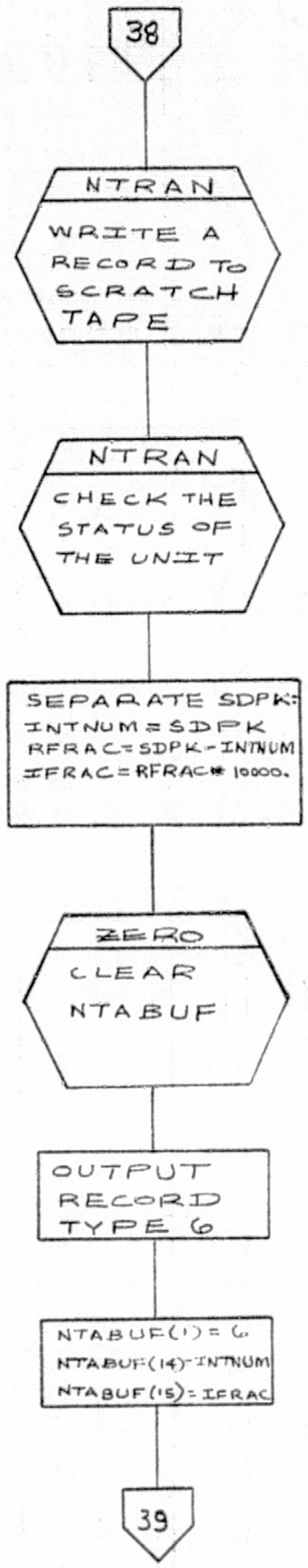

33
DDD (4596)
UPDATE THE TOTAL ACCUMULATORS
SUMTM= SUMTM + USCNTM
RMSTOT= RMSTOT+ RMSFNL
RMXSUM= RMXSUM+ MXA4
RMNSUM = RMNSUM+ MNA4
RXSMSQ= RXSMSQ+ FLOAT(MXA4)**2
RNSMSQ= RNSMSQ+ FLOAT(MNA4)**2
OMIT DEBUG PRINT ?
Y
N
EEE (4598)
DEBUG PRINT: "TOTAL ACCUMULATORS" PDUMP SUMTM, RMSTOT, RMXSUM, RMNSUM, RXSMSQ, RNSMSQ
DEBUG PRINT: PDUMP MXA4, MNA4
34

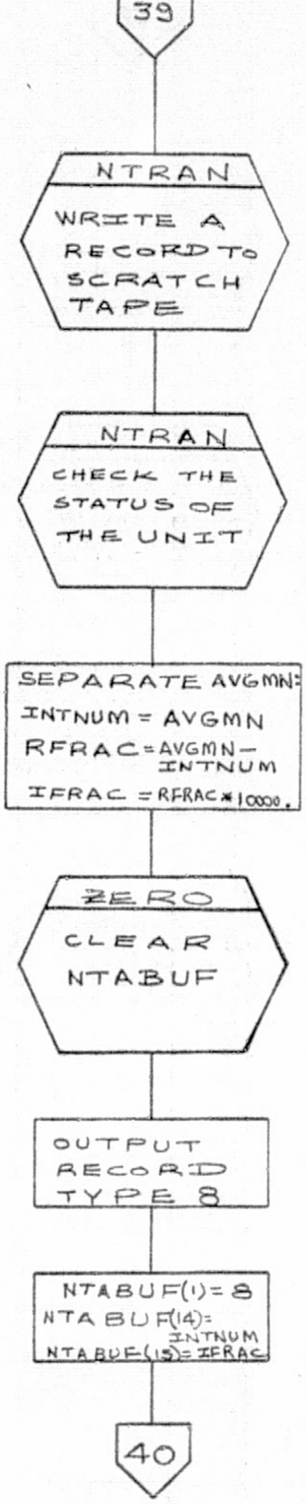

34
EEE (4598)
UPDATE VALID SCAN LINE COUNTER
NMVALD = NMVALD+1
IS THE NUMBER OF CONSECUTIVE VALID SCAN LINES UNEQUAL TO THE NUMBER REQUIRED ?
Y
FFF (4800)
N
PROCESSING FOR THE REQUESTED NUMBER OF SCANS HAS BEEN COMPLETED
COMPUTE AVERAGE OF THE SCAN LINE PEAK VALUES
COMPUTE AVERAGE OF THE SCAN LINE MINIMUM VALUES
COMPUTE STANDARD DEVIATION OF THE SCAN LINE PEAK VALUES
COMPUTE STANDARD DEVIATION OF THE SCAN LINE MINIMUM VALUES
35


35
COMPUTE AVERAGE OF THE SCAN LINE'S ROOT MEAN SQUARE VALUES
COMPUTE AVERAGE TIME OF A SCAN LINE
G (4600)
AVGMX =(RMXSUM/ NMVALD)*.005
AVGMN=(RMNSUM/ NMVALD)*.005
IS THE NUMBER OF CONSECUTIVE VALID SCANS PROCESSED UNEQUAL TO 1?
Y
N
GGG (4601)
SDPK= 0.
SDMIN = 0.
HHH (4602)


36
GGG (4601)
SDPK = .005* SQRT (((RXSMSQ* NMVALD) - (RMXSUM **2)) / (NMVALD* (NMVALD-1)))
SDMIN = .005* SQRT(((RNSMSQ* NMVALD) - (RMNSUM **2))/ (NMVALD*(NMVALD -1)))
HHH (4602)
AVGLIN = (RMSTOT/ NMVALD)* .005
AVSCTM = (SUMTM/ NMVALD)
OUTPUT:
AVG W/L RAMP PEAK (AVGMX)
AVG W/L RAMP MIN (AVGMN)
SD W/L RAMP PEAK (SDPK)
SD W/L RAMP MIN (SDMIN)
AVG SCAN RMS (AVGLIN)
AVG SCAN INTERVAL TIME (AVSCTM)
OMIT DEBUG PRINT ?
Y
N
J (4603)
37

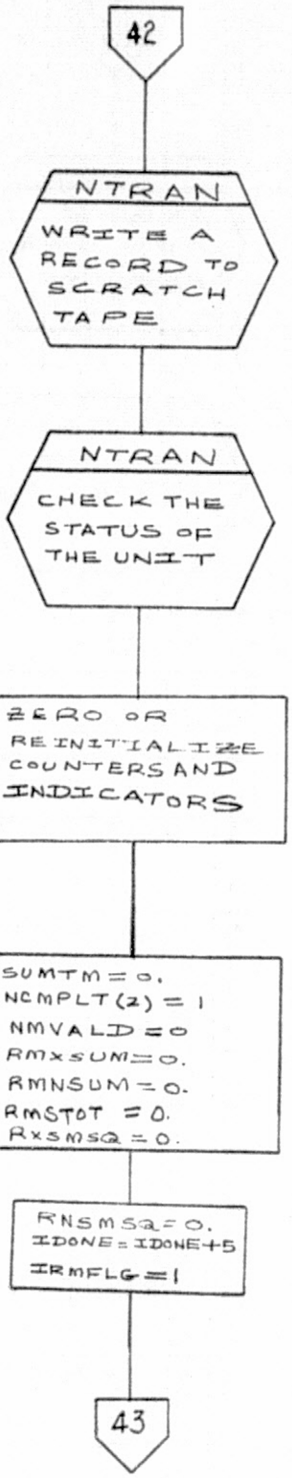

37
DEBUG PRINT: PDUMP AVGMX, AVGMN, SDPK, SDMIN, AVGLIN, AVSCTM
J
SET LOGICAL UNIT NUMBER LO = 12
SEPARATE AVGMX: INTNUM = AVGMX RFRAC = AVGMX - INTNUM IFRAC = RFRAC * 10000.
ZERO
CLEAR NTABUF
OUTPUT RECORD TYPE 4
NTABUF (1) = 4 NTABUF (14) = INTNUM NTABUF (15) = IFRAC
38

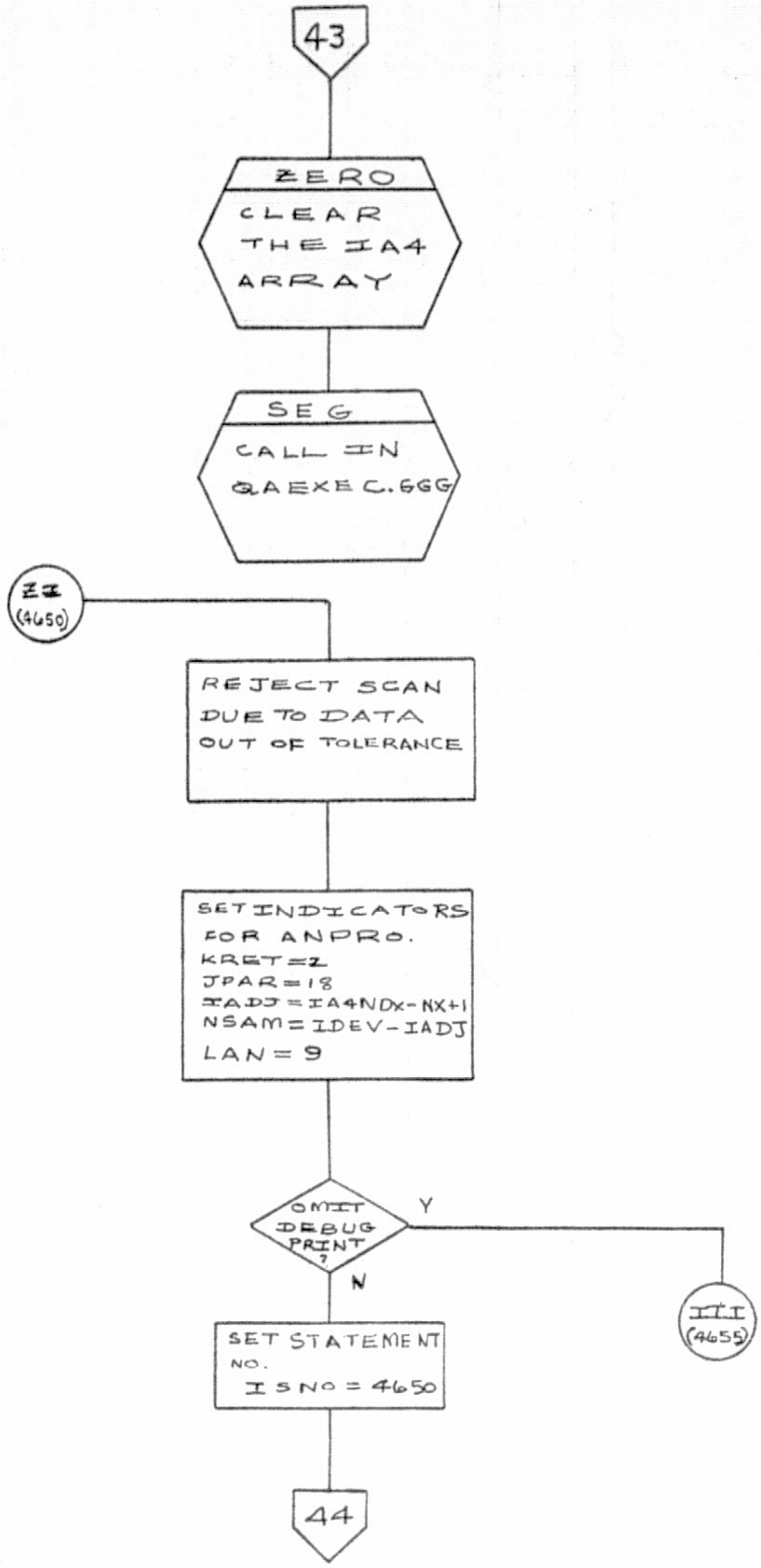

38
NTRAN
WRITE A RECORD TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
SEPARATE SDPK:
INTNUM = SDPK
RFRAC = SDPK - INTNUM
IFRAC = RFRAC * 10000.
ZERO
CLEAR NTABUF
OUTPUT RECORD TYPE 6
NTABUF(1) = 6
NTABUF(14) = INTNUM
NTABUF(15) = IFRAC
39

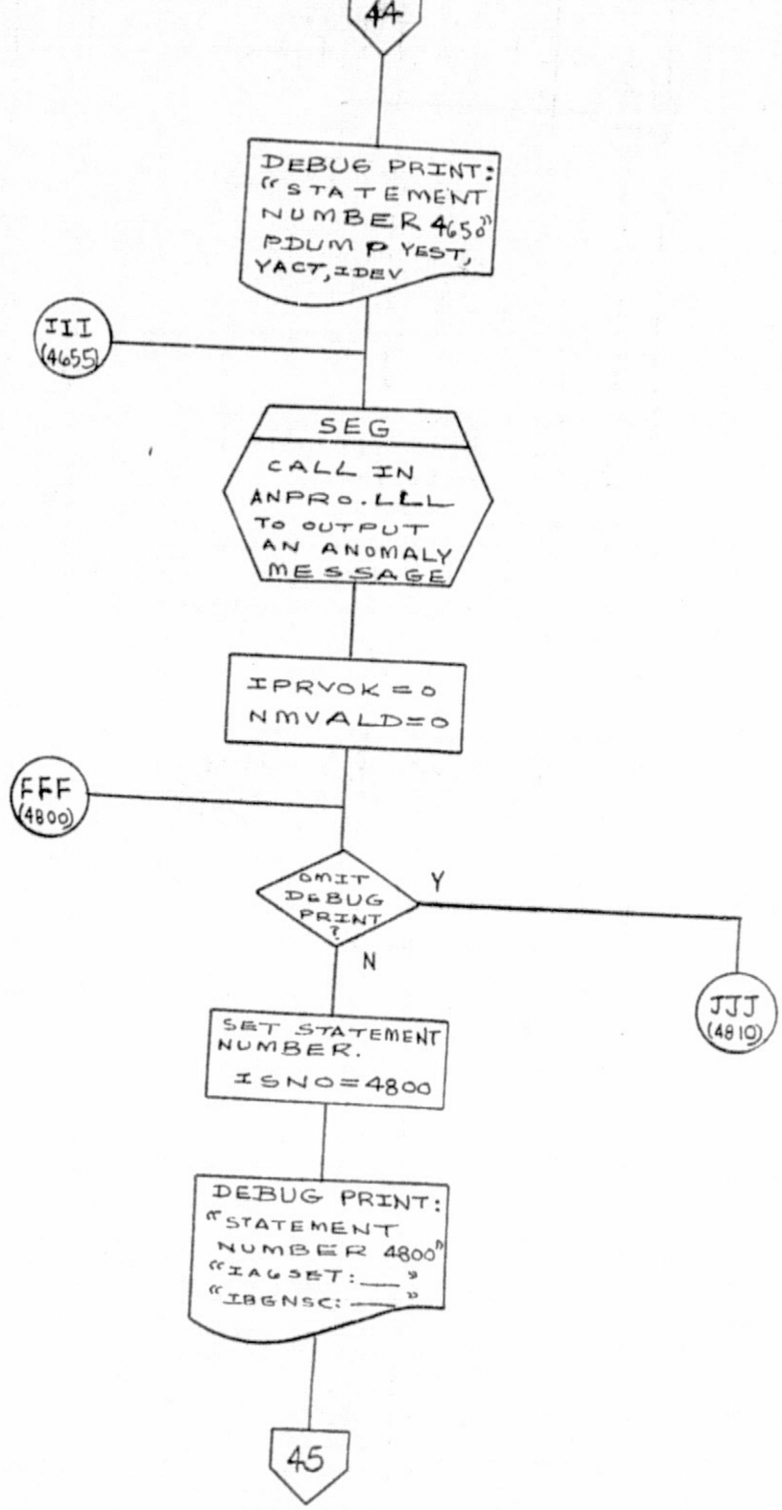

39
NTRAN
WRITE A RECORD TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
SEPARATE AVGMN:
INTNUM = AVGMN
RFRAC=AVGMN-INTNUM
IFRAC =RFRAC*10000.
ZERO
CLEAR NTABUF
OUTPUT RECORD TYPE 8
NTABUF(1)= 8
NTABUF(14)= INTNUM
NTABUF(15)= IFRAC
40

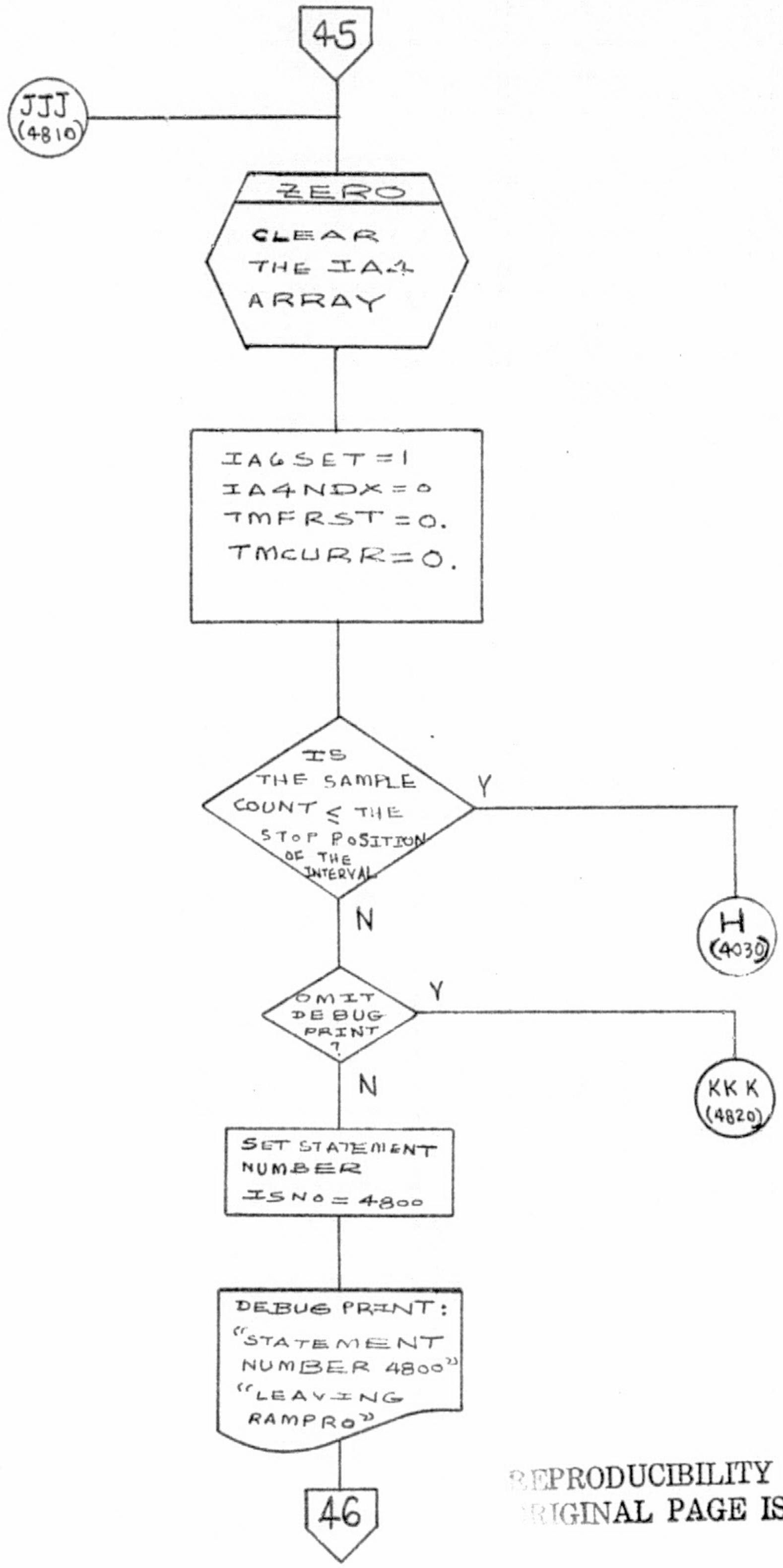

40
NTRAN
WRITE A RECORD TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
SEPARATE SDMIN:
INTNUM= SDMIN
RFRAC=SDMIN-INTNUM
IFRAC = RFRAC*10000.
ZERO
CLEAR NTABUF
OUTPUT RECORD TYPE 10
NTABUF(1) = 10
NTABUF(14)= INTNUM
NTABUF(15)= IFRAC
41

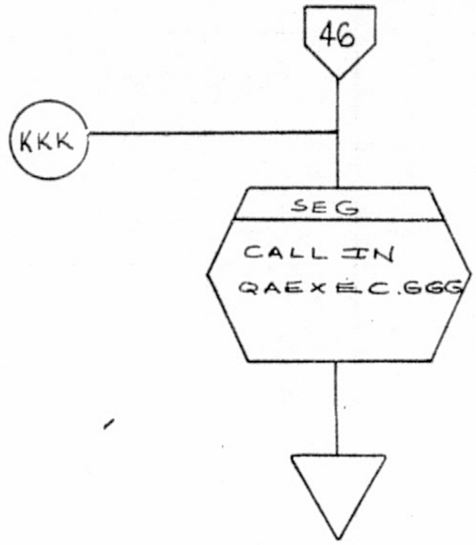

41
NTRAN
WRITE A RECORD TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
SEPARATE AVGLIN:
RFRAC=AVGLIN-INTNUM
IFRAC=RFRAC*10000.
JNTNUM=AVSCTM
XFRAC=AVSCTM-JNTNUM
JFRAC=XFRAC*10000.
ZERO
CLEAR NTABUF
OUTPUT RECORD TYPE 12
NTABUF(1)=12
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
NTABUF(16)=JNTNUM
NTABUF(17)=JFRAC
42

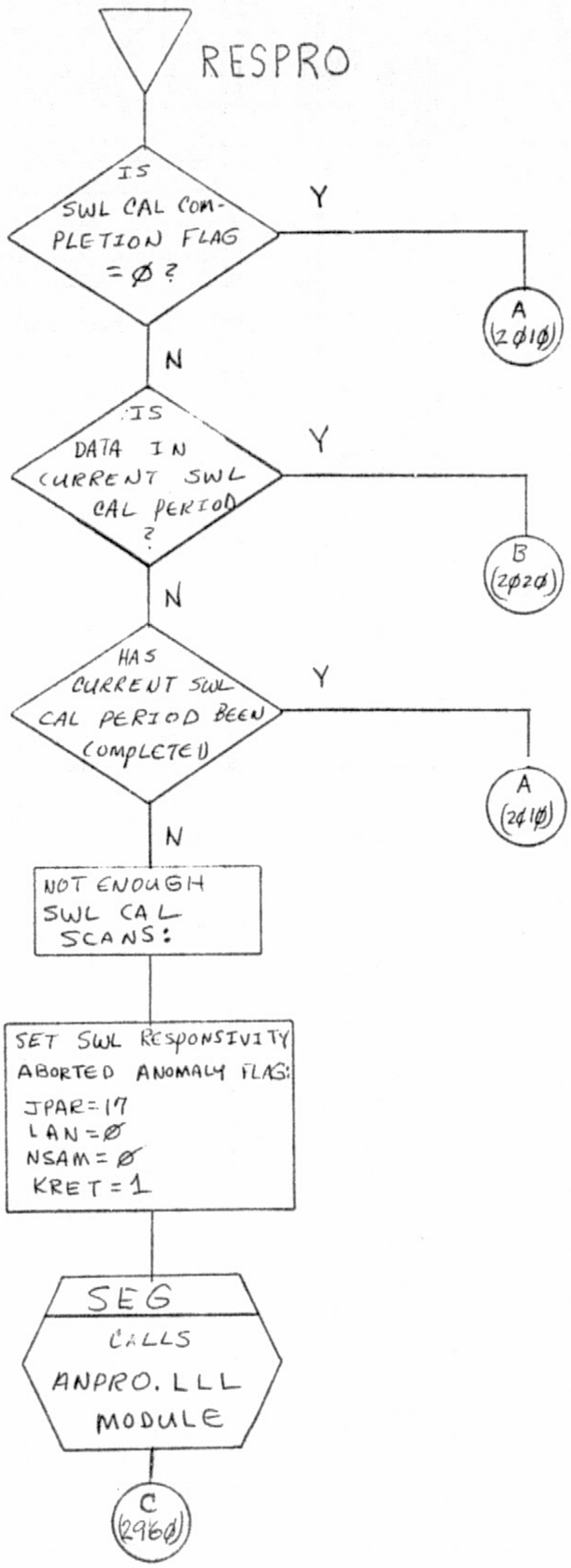

42
NTRAN
WRITE A RECORD TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
ZERO OR REINITIALIZE COUNTERS AND INDICATORS
SUMTM = 0.
NCMPLT(2) = 1
NMVALD = 0
RMXSUM = 0.
RMNSUM = 0.
RMSTOT = 0.
RXSMSQ = 0.
RNSMSQ = 0.
IDONE = IDONE+5
IRMFLG = 1
43

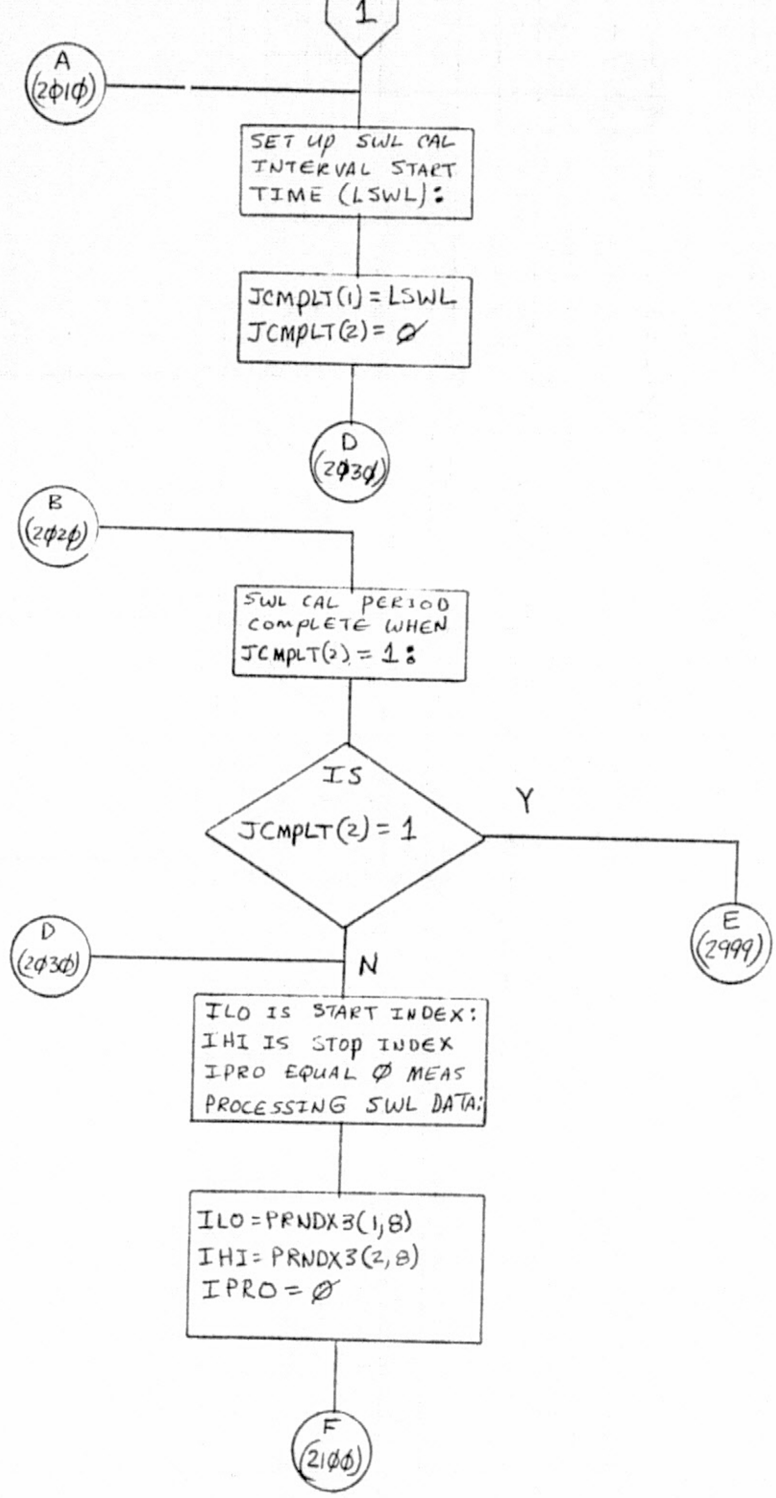

43
ZERO
CLEAR
THE IA4
ARRAY
SEG
CALL IN
QAEXEC.GGG
II
(4650)
REJECT SCAN
DUE TO DATA
OUT OF TOLERANCE
SET INDICATORS
FOR ANPRO.
KRET=2
JPAR=18
IADJ=IA4NDX-NX+1
NSAM=IDEV-IADJ
LAN=9
OMIT
DEBUG
PRINT
?
Y
N
III
(4655)
SET STATEMENT
NO.
ISNO=4650
44

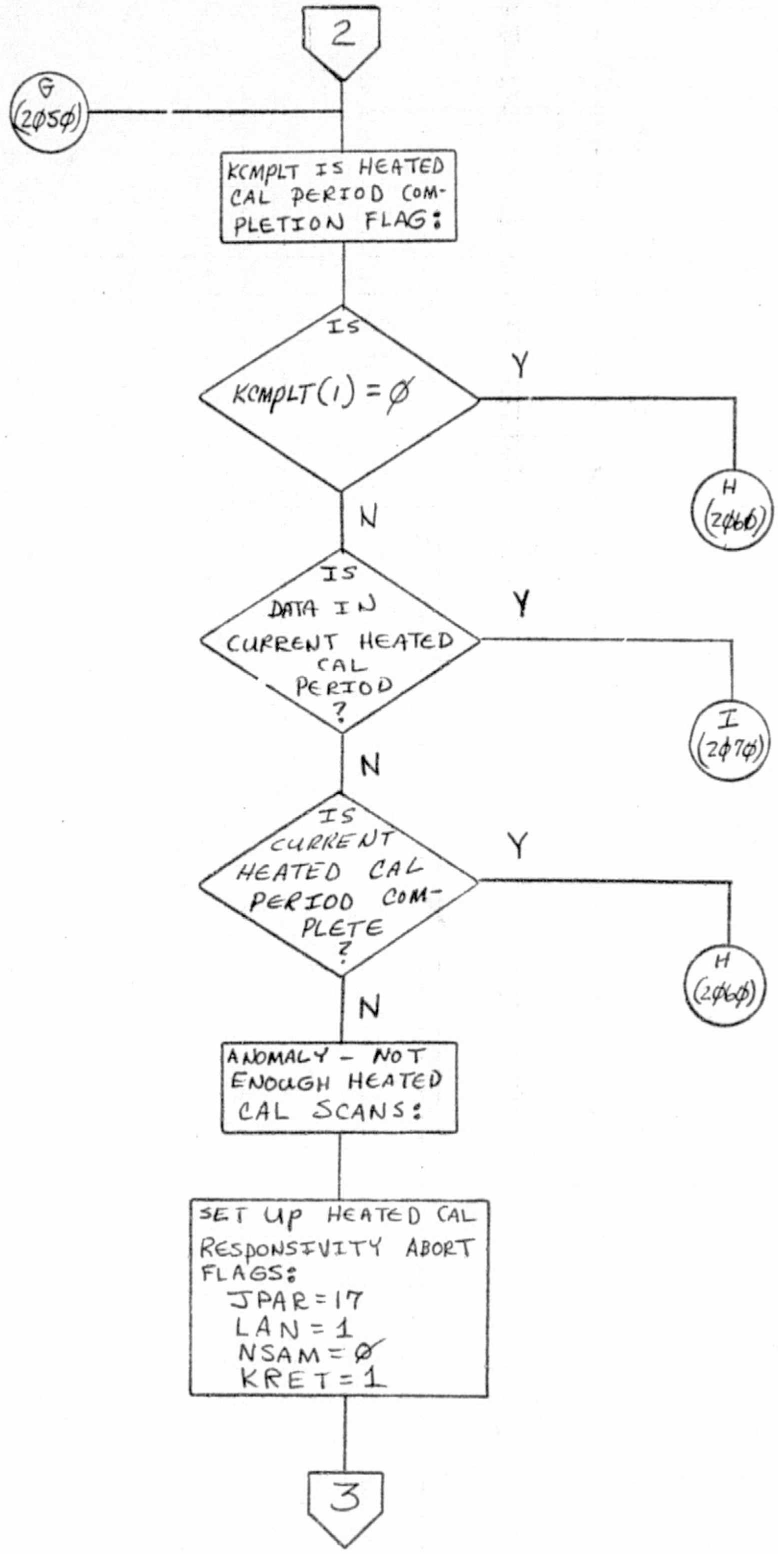

44
DEBUG PRINT: "STATEMENT NUMBER 4650" PDUMP YEST, YACT, IDEV
III (4655)
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
IPRVOK=0
NMVALD=0
FFF (4800)
OMIT DEBUG PRINT ?
Y
N
JJJ (4810)
SET STATEMENT NUMBER.
ISNO=4800
DEBUG PRINT: "STATEMENT NUMBER 4800" "IAGSET: ___" "IBGNSC: ___"
45

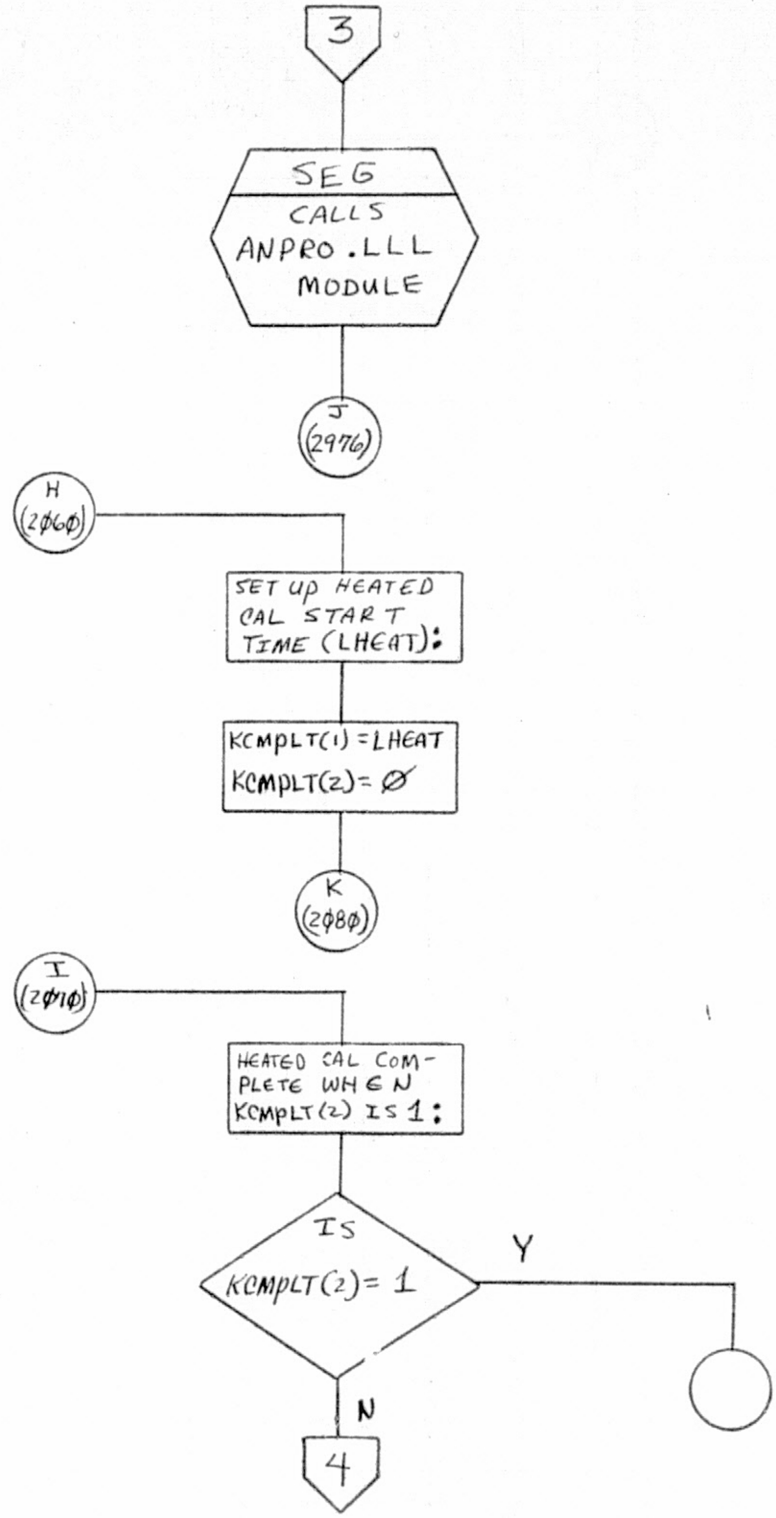

45
JJJ
(4810)
ZERO
CLEAR THE IA4 ARRAY
IA6SET=1
IA4NDX=0
TMFRST=0.
TMCURR=0.
IS THE SAMPLE COUNT ≤ THE STOP POSITION OF THE INTERVAL
Y
H
(4030)
N
OMIT DEBUG PRINT ?
Y
KKK
(4820)
N
SET STATEMENT NUMBER ISNO=4800
DEBUG PRINT: "STATEMENT NUMBER 4800" "LEAVING RAMPRO"
46

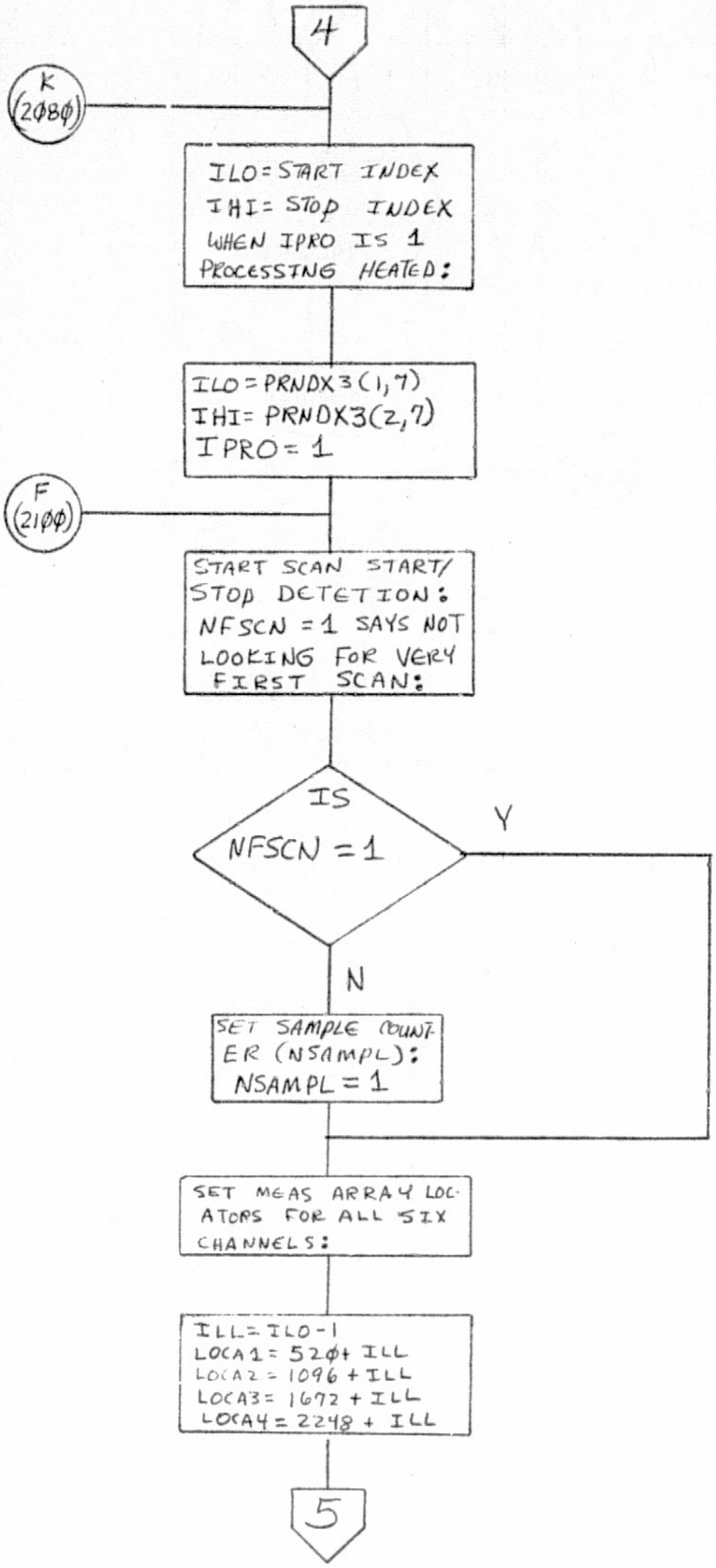

46
KKK
SEG
CALL IN
QAEXEC.GGG

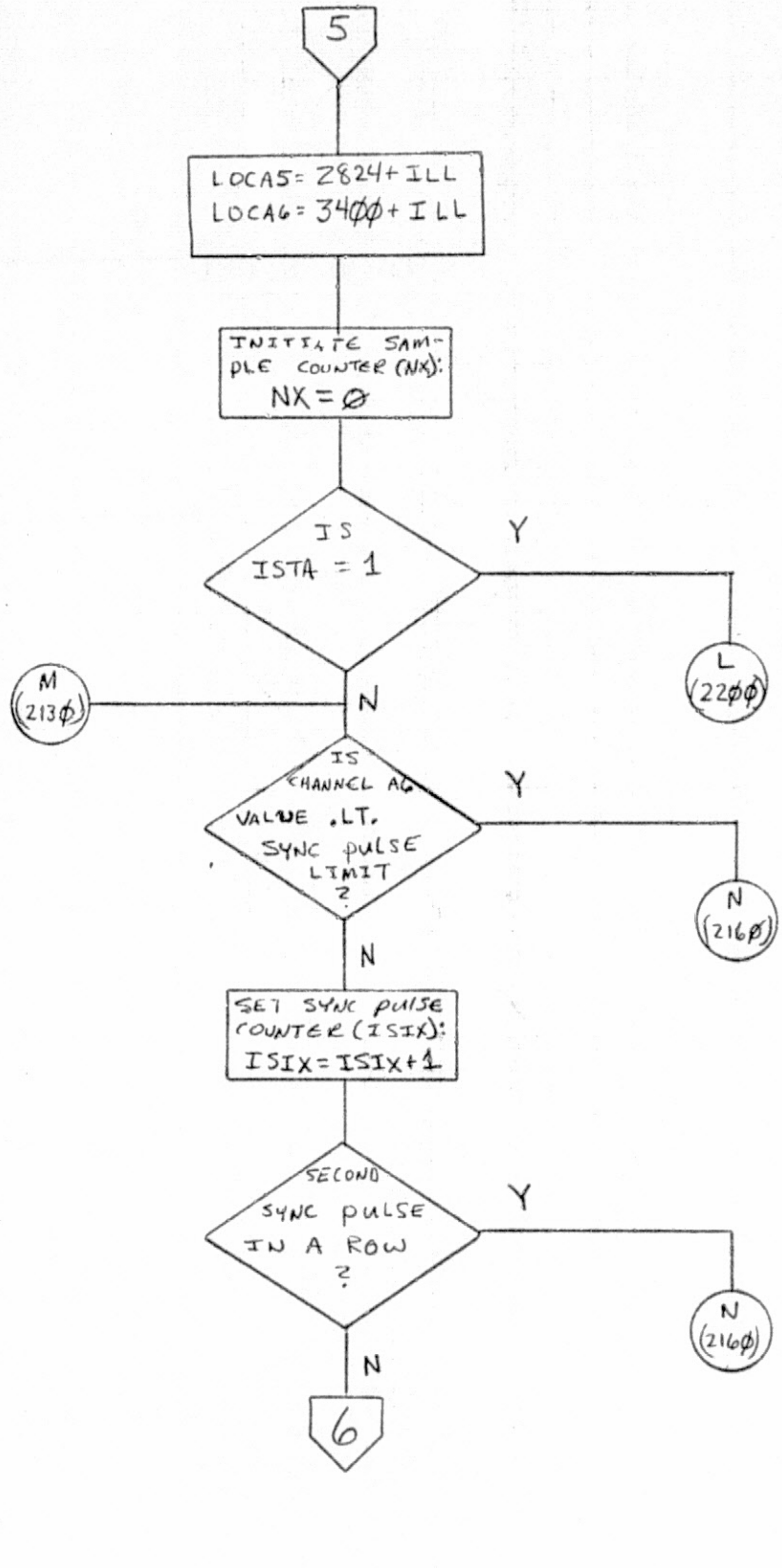

RESPRO
IS SWL CAL COMPLETION FLAG = Ø ?
Y
A (2Ø1Ø)
N
IS DATA IN CURRENT SWL CAL PERIOD ?
Y
B (2Ø2Ø)
N
HAS CURRENT SWL CAL PERIOD BEEN COMPLETED
Y
A (2Ø1Ø)
N
NOT ENOUGH SWL CAL SCANS:
SET SWL RESPONSIVITY ABORTED ANOMALY FLAG:
JPAR=17
LAN=Ø
NSAM=Ø
KRET=1
SEG
CALLS ANPRO.LLL MODULE
C (296Ø)

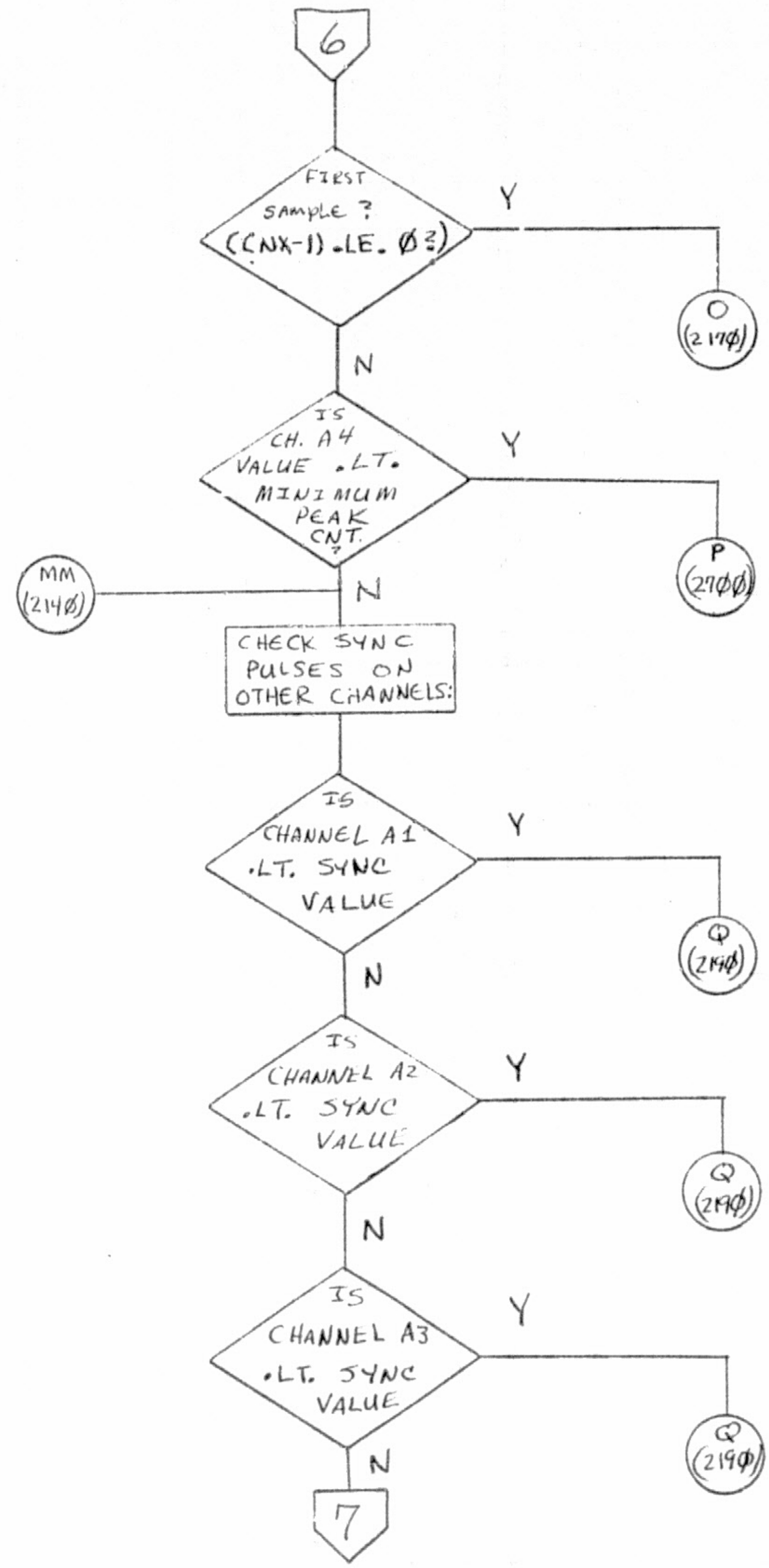

1
A (2Ø1Ø)
SET UP SWL CAL INTERVAL START TIME (LSWL):
JCMPLT(1) = LSWL
JCMPLT(2) = Ø
D (2Ø3Ø)
B (2Ø2Ø)
SWL CAL PERIOD COMPLETE WHEN JCMPLT(2) = 1:
IS JCMPLT(2) = 1
Y
E (2999)
D (2Ø3Ø)
N
ILO IS START INDEX: IHI IS STOP INDEX IPRO EQUAL Ø MEAS PROCESSING SWL DATA:
ILO = PRNDX3(1,8)
IHI = PRNDX3(2,8)
IPRO = Ø
F (21ØØ)

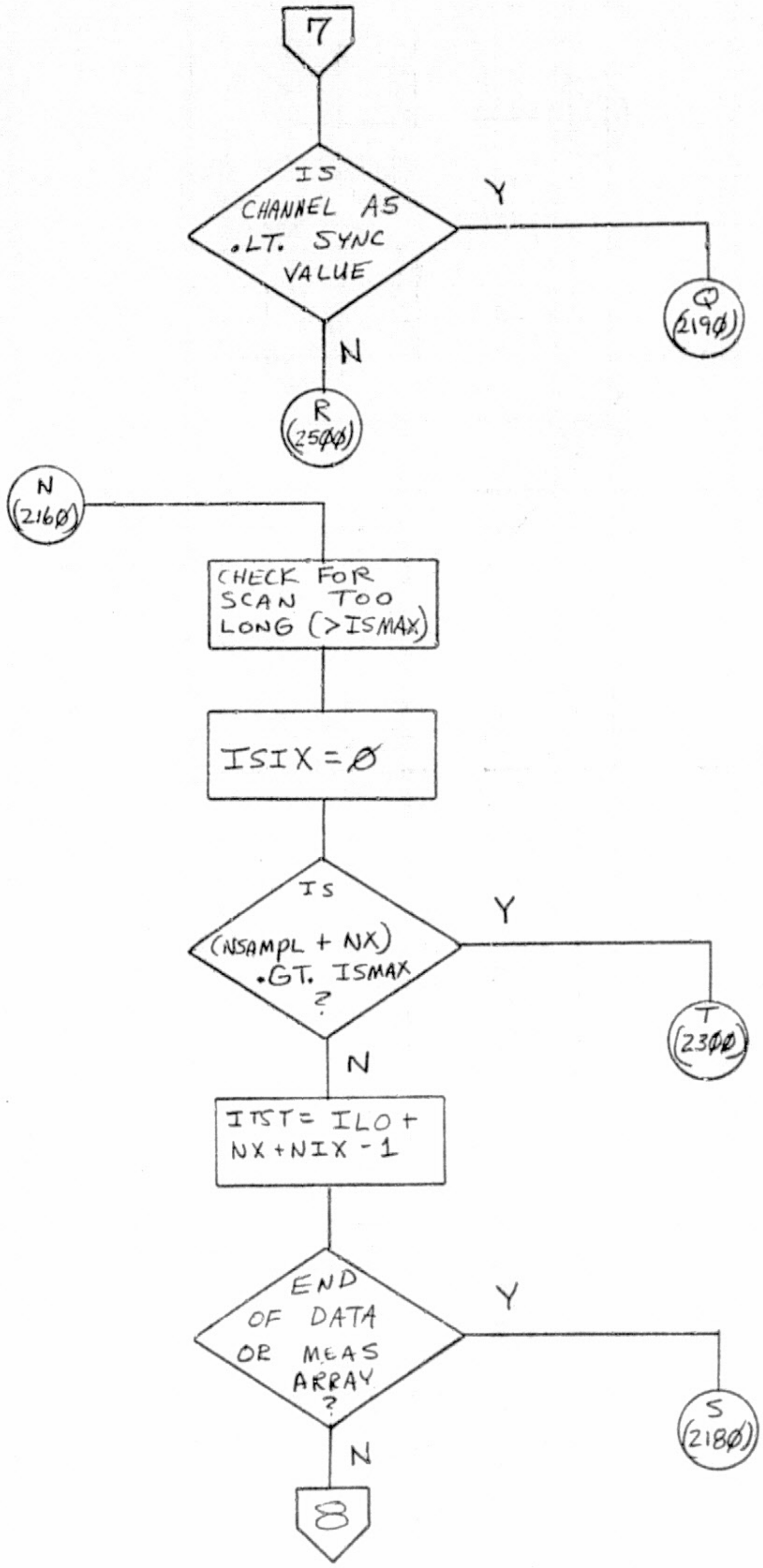

2
G
(2Ø5Ø)
KCMPLT IS HEATED CAL PERIOD COMPLETION FLAG:
IS KCMPLT(1) = Ø
Y
H
(2Ø6Ø)
N
IS DATA IN CURRENT HEATED CAL PERIOD ?
Y
I
(2Ø7Ø)
N
IS CURRENT HEATED CAL PERIOD COMPLETE ?
Y
H
(2Ø6Ø)
N
ANOMALY - NOT ENOUGH HEATED CAL SCANS:
SET UP HEATED CAL RESPONSIVITY ABORT FLAGS:
JPAR=17
LAN=1
NSAM=Ø
KRET=1
3

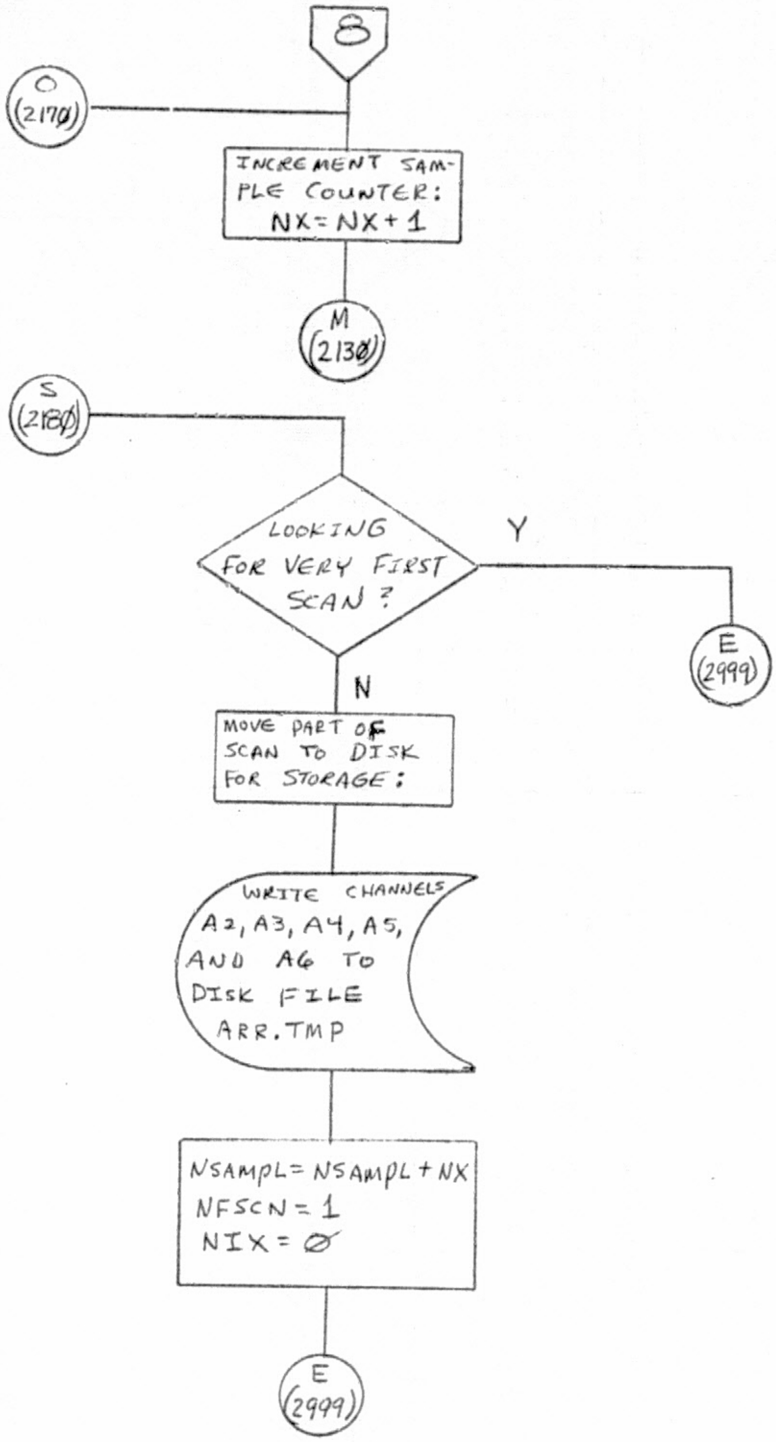

3
SEG
CALLS
ANPRO.LLL
MODULE
J
(2976)
H
(2Ø6Ø)
SET UP HEATED
CAL START
TIME (LHEAT):
KCMPLT(1) = LHEAT
KCMPLT(2) = Ø
K
(2Ø8Ø)
I
(2Ø7Ø)
HEATED CAL COM-
PLETE WHEN
KCMPLT(2) IS 1:
IS
KCMPLT(2) = 1
Y
N
4

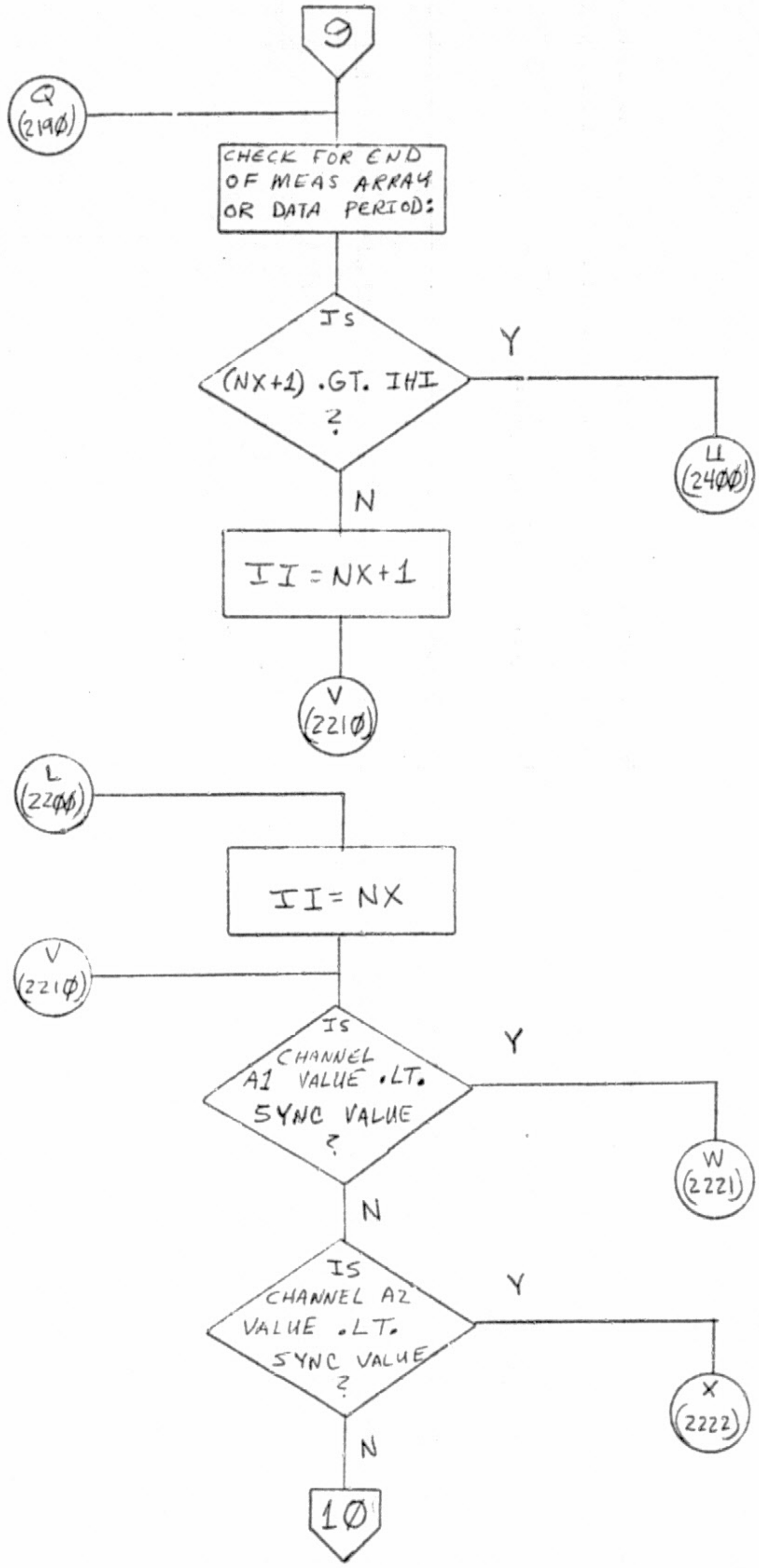

4
K
(2Ø8Ø)
ILO=START INDEX
IHI= STOP INDEX
WHEN IPRO IS 1
PROCESSING HEATED:
ILO=PRNDX3(1,7)
IHI= PRNDX3(2,7)
IPRO=1
F
(21ØØ)
START SCAN START/
STOP DETETION:
NFSCN =1 SAYS NOT
LOOKING FOR VERY
FIRST SCAN:
IS
NFSCN =1
Y
N
SET SAMPLE COUNT-
ER (NSAMPL):
NSAMPL=1
SET MEAS ARRAY LOC-
ATORS FOR ALL SIX
CHANNELS:
ILL=ILO-1
LOCA1=52Ø+ ILL
LOCA2=1096 + ILL
LOCA3=1672 + ILL
LOCA4=2248 + ILL
5

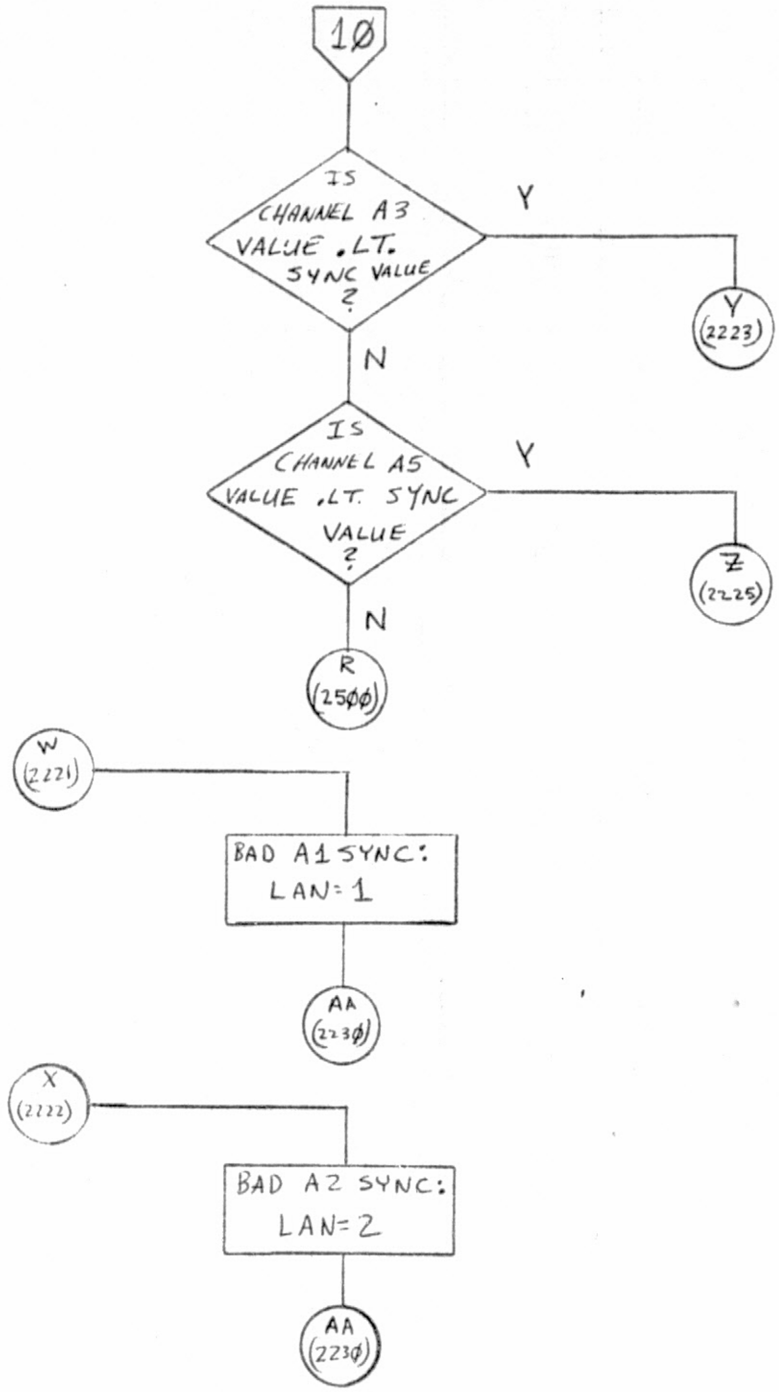

5
LOCA5 = 2824 + ILL
LOCA6 = 3400 + ILL
INITIATE SAMPLE COUNTER (NX):
NX = 0
IS ISTA = 1
Y
L (2200)
M (2130)
N
IS CHANNEL A6 VALUE .LT. SYNC PULSE LIMIT ?
Y
N (2160)
N
SET SYNC PULSE COUNTER (ISIX):
ISIX = ISIX + 1
SECOND SYNC PULSE IN A ROW ?
Y
N (2160)
N
6

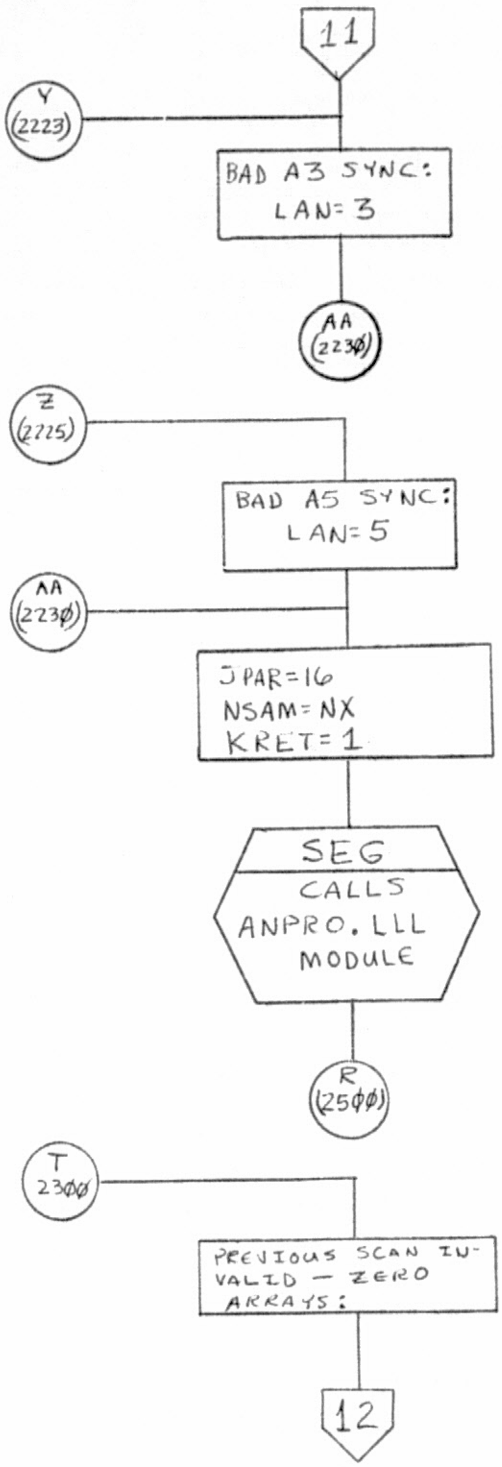

6
FIRST SAMPLE ? ((NX-1) .LE. Ø?)
Y
O (217Ø)
N
IS CH. A4 VALUE .LT. MINIMUM PEAK CNT. ?
Y
P (27ØØ)
MM (214Ø)
N
CHECK SYNC PULSES ON OTHER CHANNELS:
IS CHANNEL A1 .LT. SYNC VALUE
Y
Q (219Ø)
N
IS CHANNEL A2 .LT. SYNC VALUE
Y
Q (219Ø)
N
IS CHANNEL A3 .LT. SYNC VALUE
Y
Q (219Ø)
N
7

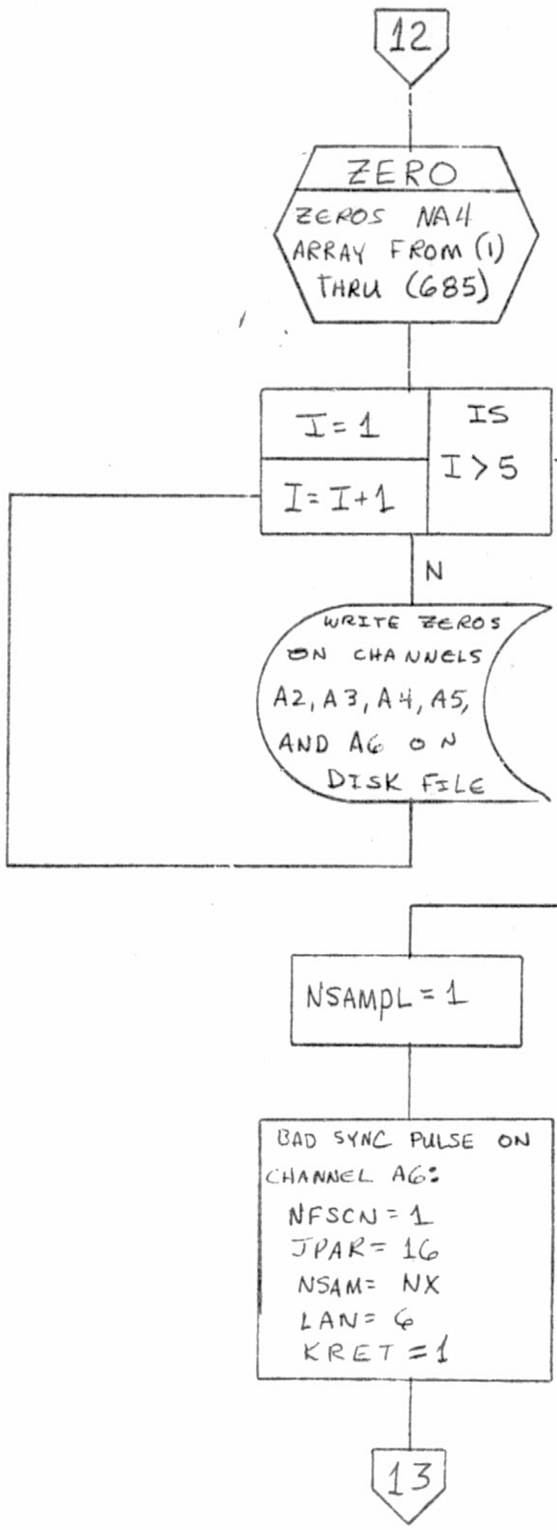

7
IS CHANNEL A5 .LT. SYNC VALUE
Y
Q (219Ø)
N
R (25ØØ)
N (216Ø)
CHECK FOR SCAN TOO LONG (>ISMAX)
ISIX = Ø
IS (NSAMPL + NX) .GT. ISMAX ?
Y
T (23ØØ)
N
ITST = ILO + NX + NIX - 1
END OF DATA OR MEAS ARRAY ?
Y
S (218Ø)
N
8

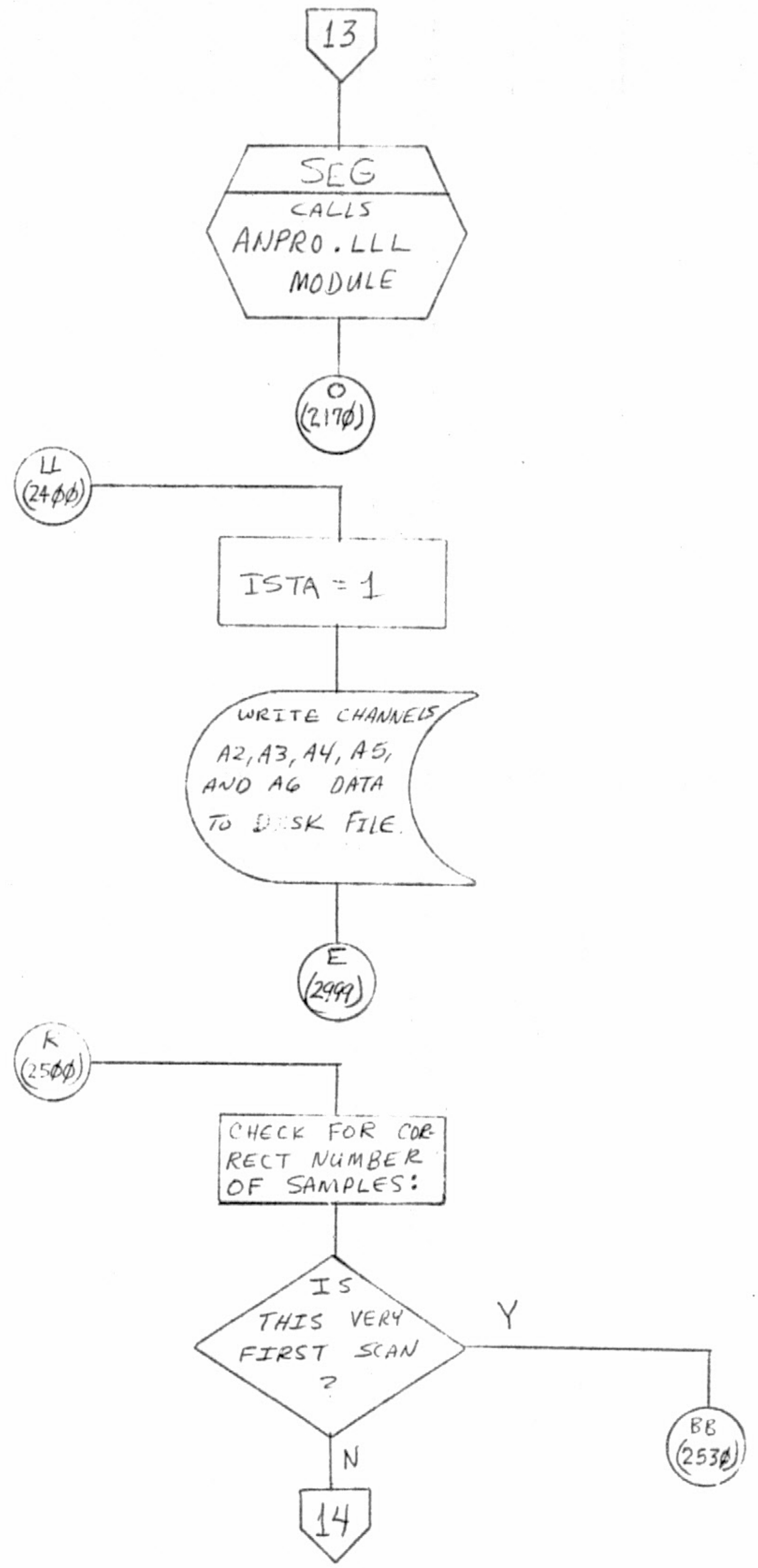

8
O
(2170)
INCREMENT SAMPLE COUNTER:
NX=NX+1
M
(2130)
S
(2180)
LOOKING FOR VERY FIRST SCAN?
Y
E
(2999)
N
MOVE PART OF SCAN TO DISK FOR STORAGE:
WRITE CHANNELS A2, A3, A4, A5, AND A6 TO DISK FILE ARR.TMP
NSAMPL=NSAMPL+NX
NFSCN=1
NIX=0
E
(2999)

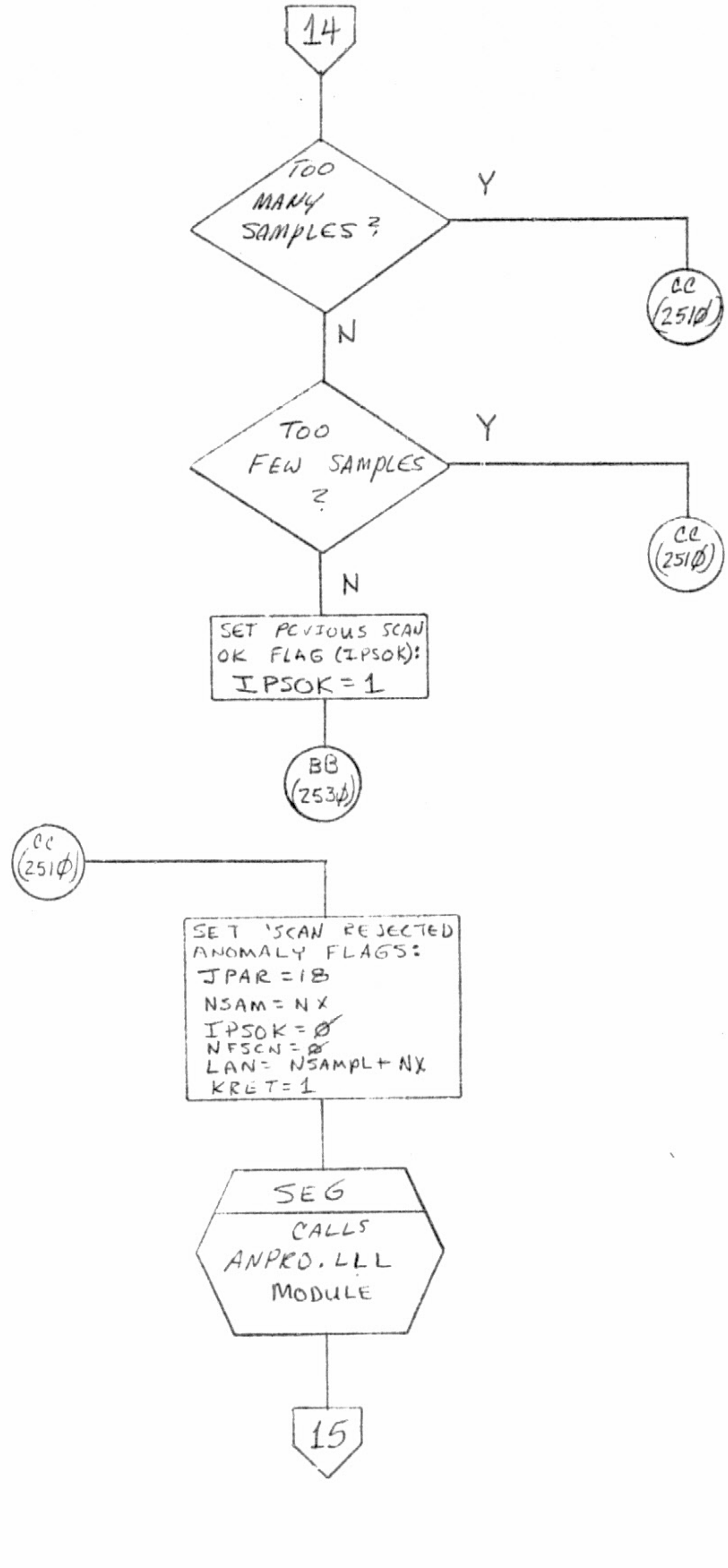

9
Q (219Ø)
CHECK FOR END OF MEAS ARRAY OR DATA PERIOD:
IS (NX+1) .GT. IHI ?
Y
U (24ØØ)
N
II = NX+1
V (221Ø)
L (22ØØ)
II = NX
V (221Ø)
IS CHANNEL A1 VALUE .LT. SYNC VALUE ?
Y
W (2221)
N
IS CHANNEL A2 VALUE .LT. SYNC VALUE ?
Y
X (2222)
N
1Ø

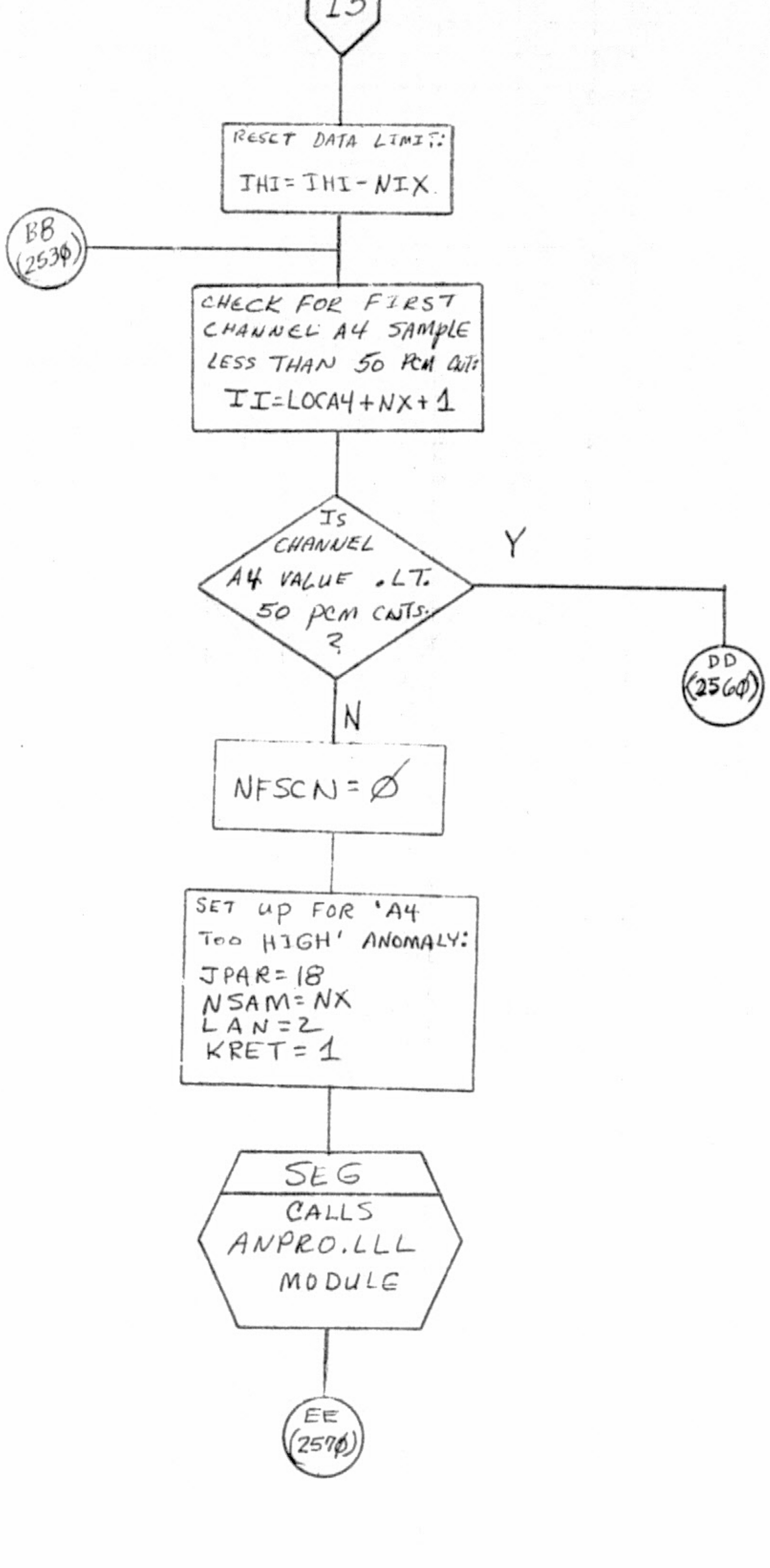

10
IS CHANNEL A3 VALUE .LT. SYNC VALUE ?
Y
Y (2223)
N
IS CHANNEL A5 VALUE .LT. SYNC VALUE ?
Y
Z (2225)
N
R (25ØØ)
W (2221)
BAD A1 SYNC: LAN=1
AA (223Ø)
X (2222)
BAD A2 SYNC: LAN=2
AA (223Ø)

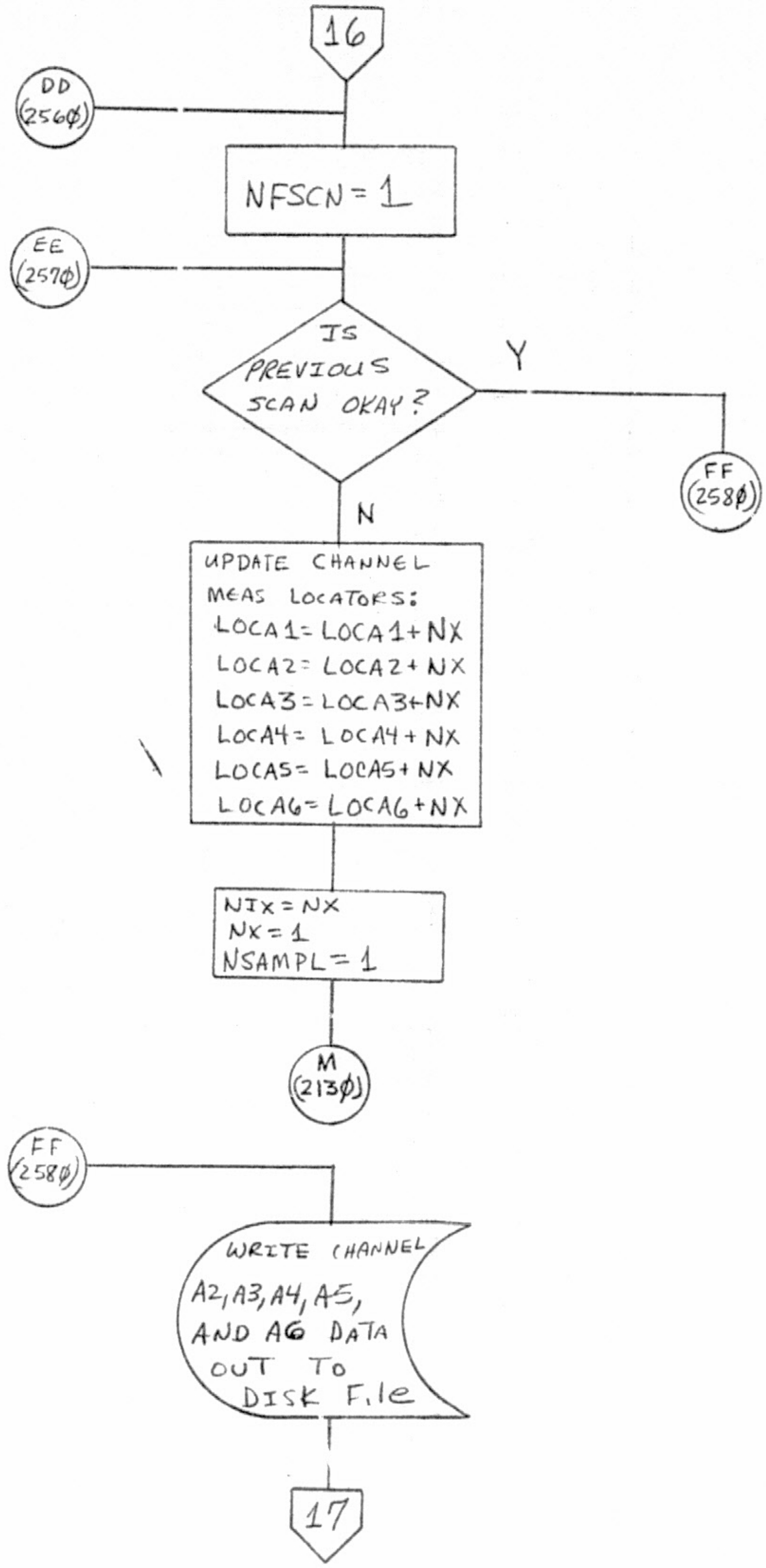

11
Y
(2223)
BAD A3 SYNC:
LAN=3
AA
(223Ø)
Z
(2225)
BAD A5 SYNC:
LAN=5
AA
(223Ø)
JPAR=16
NSAM=NX
KRET=1
SEG
CALLS
ANPRO.LLL
MODULE
R
(25ØØ)
T
23ØØ
PREVIOUS SCAN INVALID — ZERO ARRAYS:
12

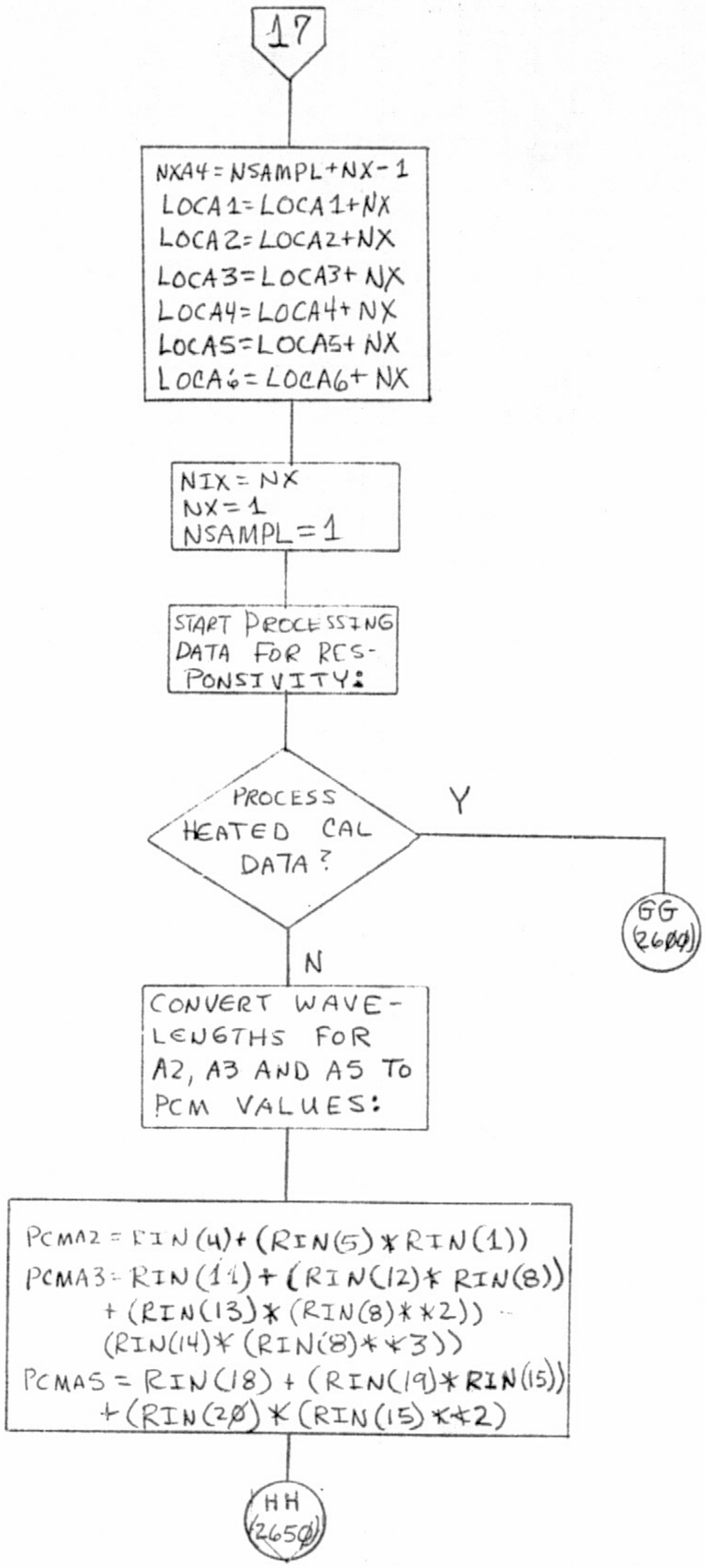

12
ZERO
ZEROS NA4 ARRAY FROM (1) THRU (685)
I=1
I=I+1
IS I>5
Y
N
WRITE ZEROS ON CHANNELS A2, A3, A4, A5, AND A6 ON DISK FILE
NSAMPL = 1
BAD SYNC PULSE ON CHANNEL A6:
NFSCN = 1
JPAR = 16
NSAM = NX
LAN = 6
KRET = 1
13

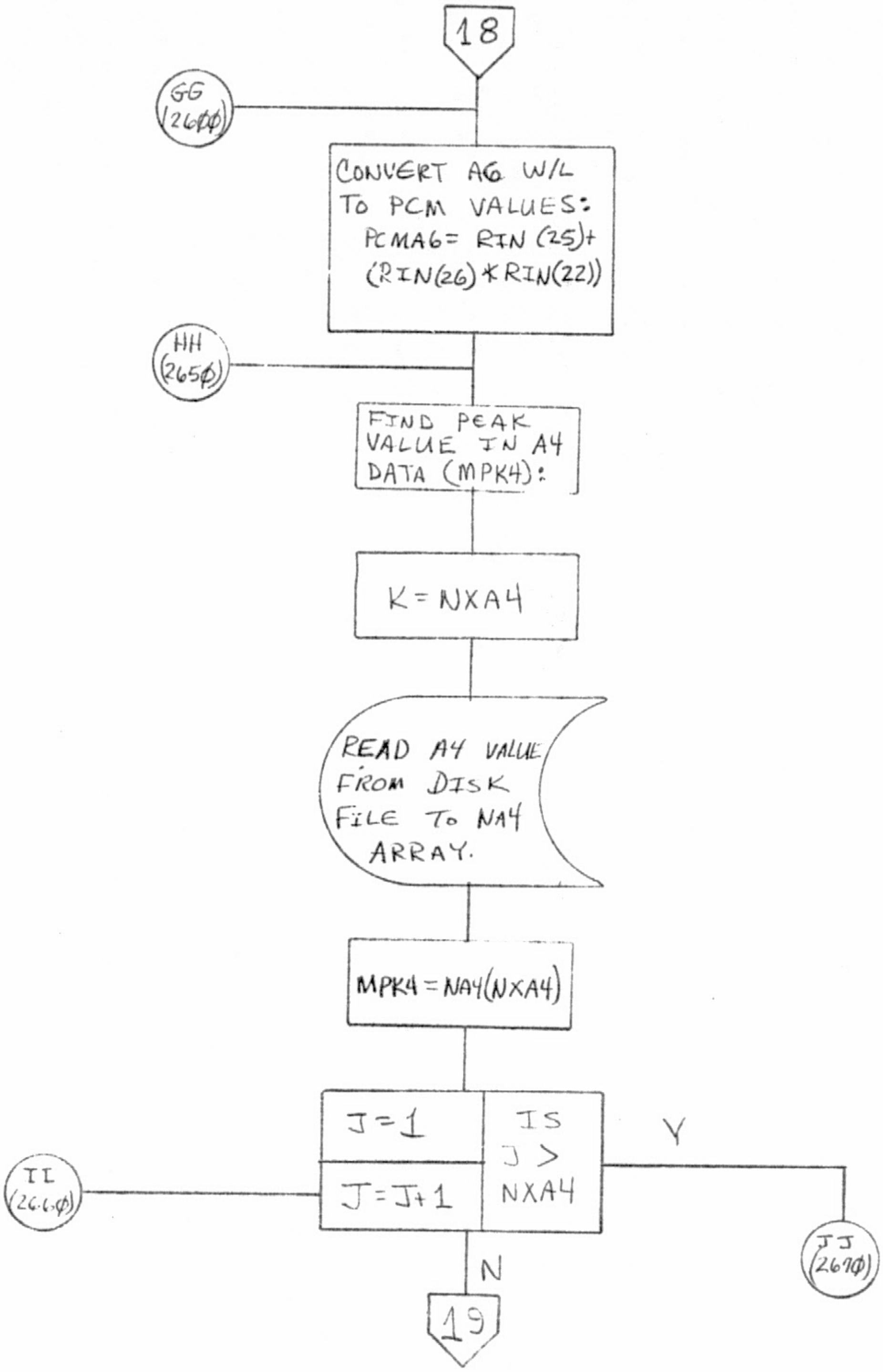

13
SEG
CALLS ANPRO.LLL MODULE
O (217Ø)
LL (24ØØ)
ISTA = 1
WRITE CHANNELS A2, A3, A4, A5, AND A6 DATA TO DISK FILE.
E (2999)
K (25ØØ)
CHECK FOR CORRECT NUMBER OF SAMPLES:
IS THIS VERY FIRST SCAN ?
Y
BB (253Ø)
N
14

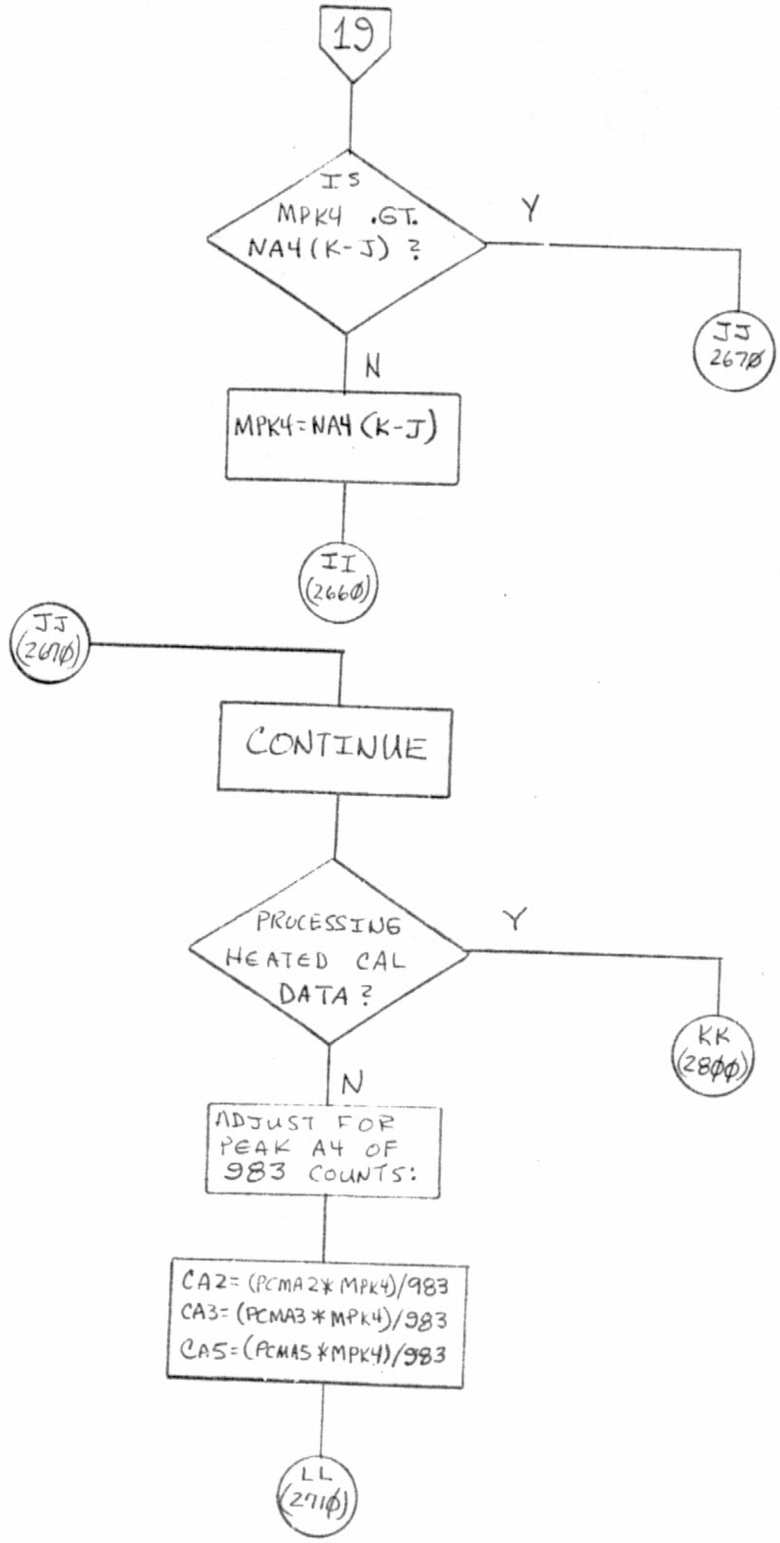

14
TOO MANY SAMPLES?
Y
CC (251∅)
N
TOO FEW SAMPLES ?
Y
CC (251∅)
N
SET PREVIOUS SCAN OK FLAG (IPSOK): IPSOK = 1
BB (253∅)
CC (251∅)
SET 'SCAN REJECTED ANOMALY FLAGS:
JPAR = 18
NSAM = NX
IPSOK = ∅
NFSCN = ∅
LAN = NSAMPL + NX
KRET = 1
SEG
CALLS ANPRO.LLL MODULE
15

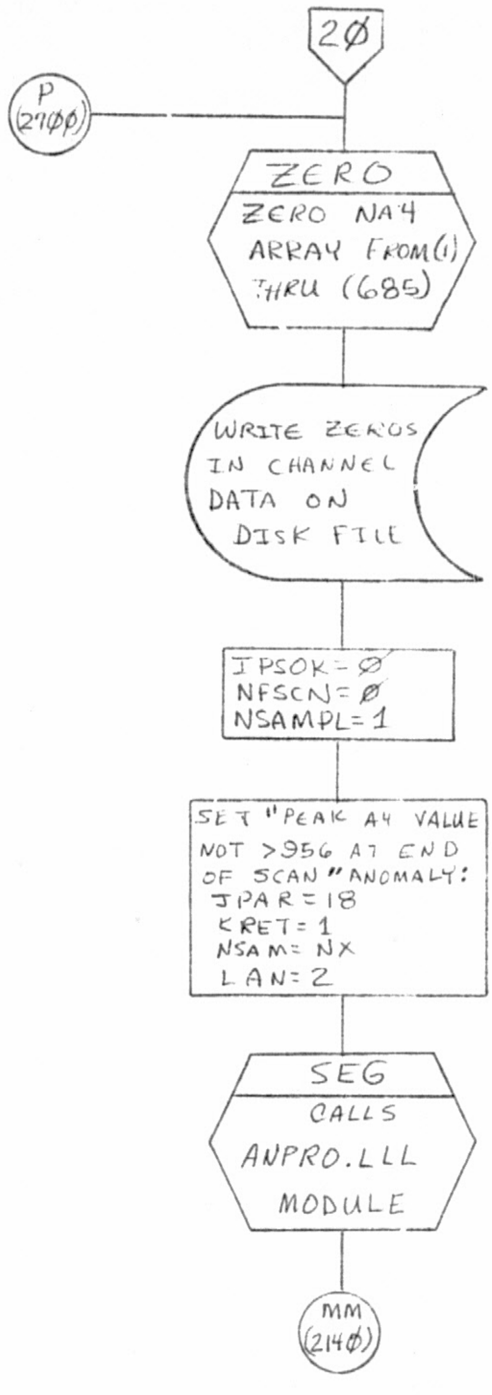

15
RESET DATA LIMIT:
IHI=IHI-NIX
BB (253∅)
CHECK FOR FIRST CHANNEL A4 SAMPLE LESS THAN 50 PCM CNTS:
II=LOCA4+NX+1
IS CHANNEL A4 VALUE .LT. 50 PCM CNTS. ?
Y
DD (256∅)
N
NFSCN=∅
SET UP FOR 'A4 TOO HIGH' ANOMALY:
JPAR=18
NSAM=NX
LAN=2
KRET=1
SEG
CALLS ANPRO.LLL MODULE
EE (257∅)

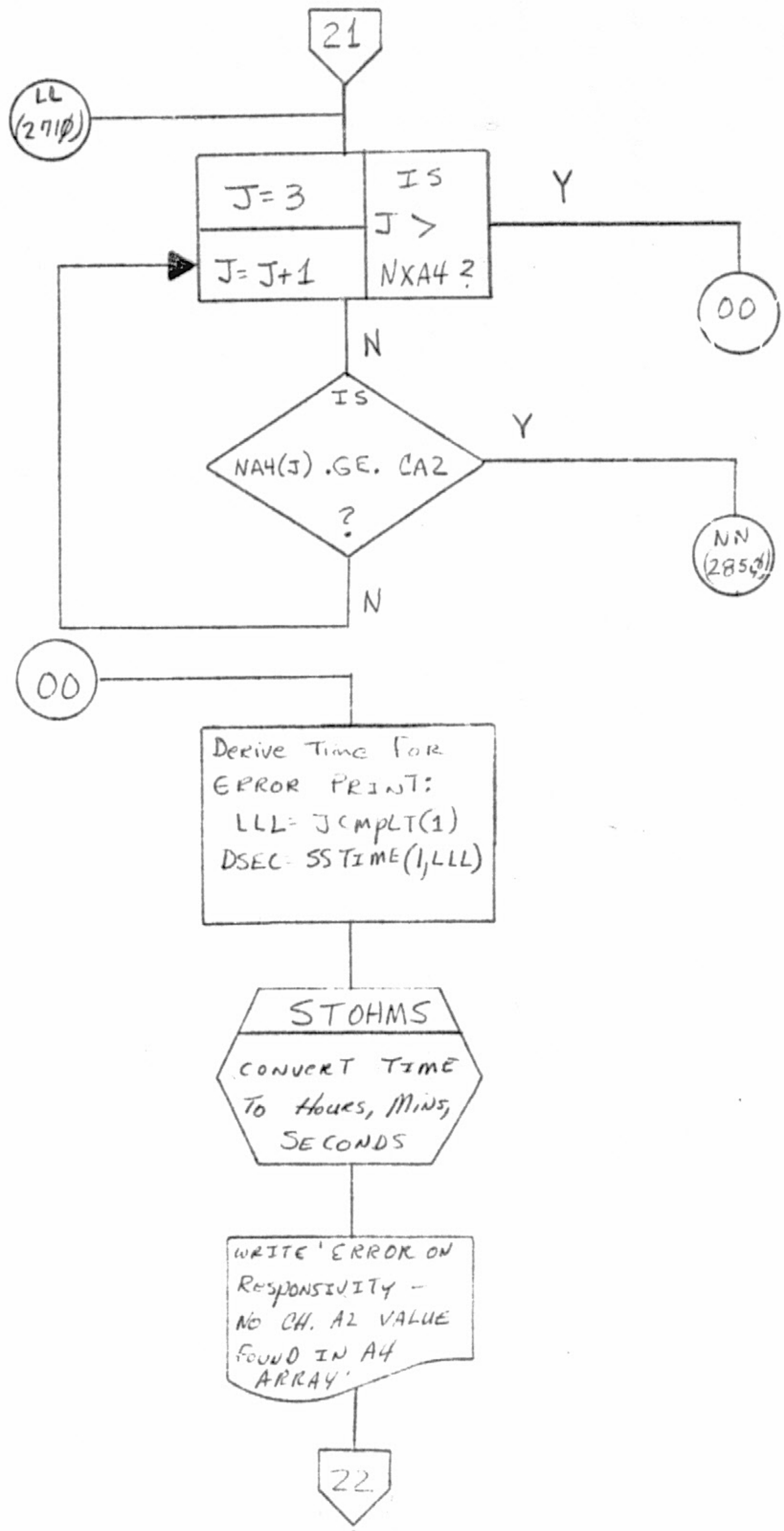

16
DD (2560)
NFSCN = 1
EE (2570)
IS PREVIOUS SCAN OKAY?
Y
FF (2580)
N
UPDATE CHANNEL MEAS LOCATORS:
LOCA1 = LOCA1 + NX
LOCA2 = LOCA2 + NX
LOCA3 = LOCA3 + NX
LOCA4 = LOCA4 + NX
LOCA5 = LOCA5 + NX
LOCA6 = LOCA6 + NX
NIX = NX
NX = 1
NSAMPL = 1
M (2130)
FF (2580)
WRITE CHANNEL A2, A3, A4, A5, AND A6 DATA OUT TO DISK File
17

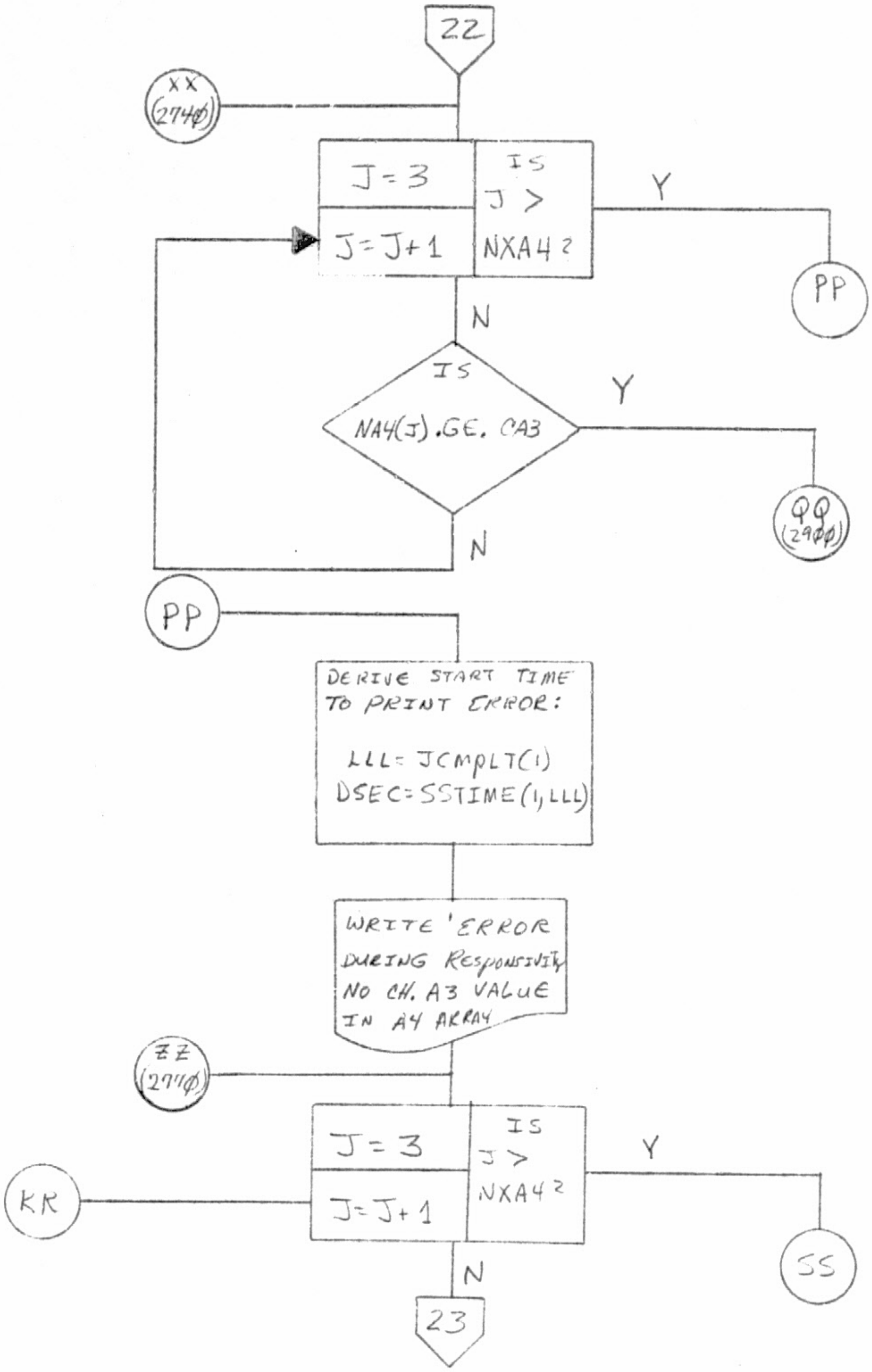

17
NXA4=NSAMPL+NX-1
LOCA1=LOCA1+NX
LOCA2=LOCA2+NX
LOCA3=LOCA3+NX
LOCA4=LOCA4+NX
LOCA5=LOCA5+NX
LOCA6=LOCA6+NX
NIX=NX
NX=1
NSAMPL=1
START PROCESSING DATA FOR RES-PONSIVITY:
PROCESS HEATED CAL DATA?
Y
GG (26ØØ)
N
CONVERT WAVE-LENGTHS FOR A2, A3 AND A5 TO PCM VALUES:
PCMA2=RIN(4)+(RIN(5)*RIN(1))
PCMA3=RIN(11)+(RIN(12)*RIN(8))
+(RIN(13)*(RIN(8)**2))
(RIN(14)*(RIN(8)**3))
PCMA5=RIN(18)+(RIN(19)*RIN(15))
+(RIN(2Ø)*(RIN(15)**2)
HH (265Ø)

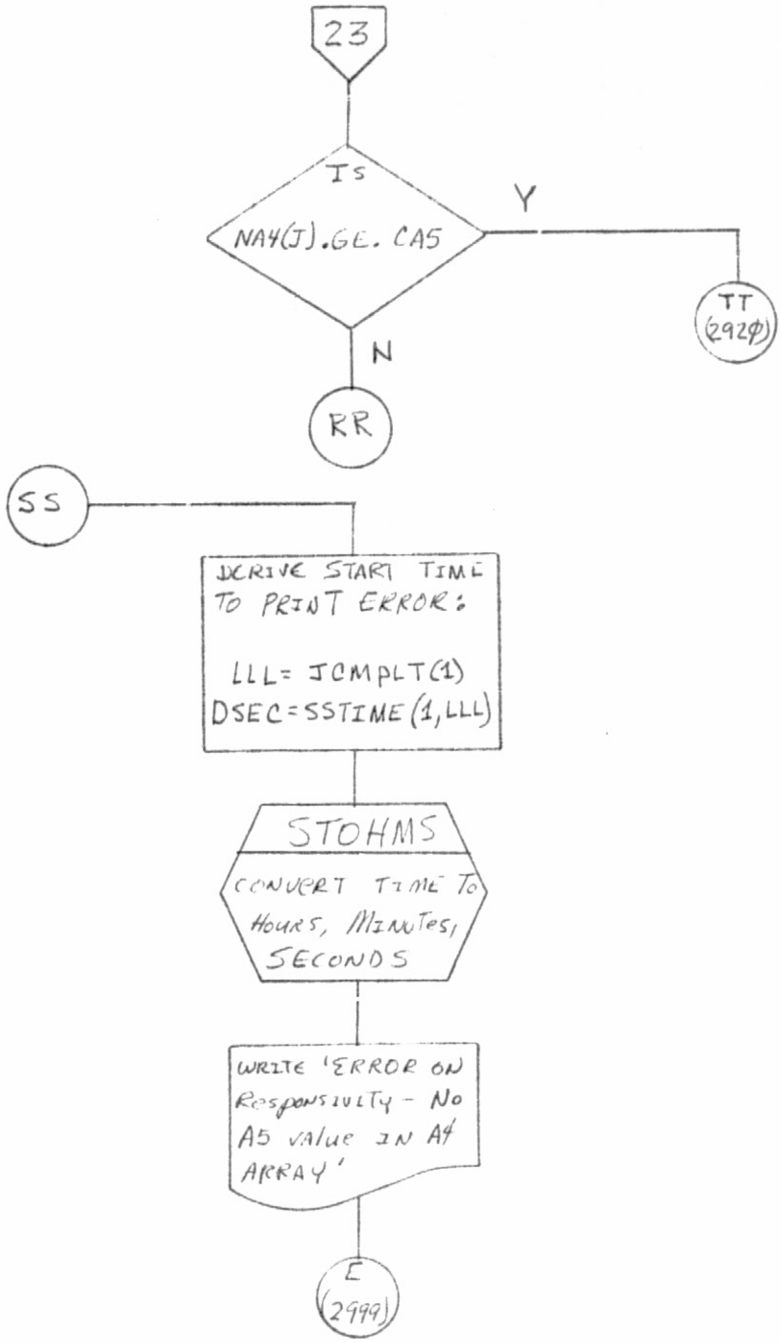

18
GG (26ØØ)
CONVERT A6 W/L TO PCM VALUES:
PCMA6 = RIN (25)+ (RIN(26) * RIN(22))
HH (265Ø)
FIND PEAK VALUE IN A4 DATA (MPK4):
K = NXA4
READ A4 VALUE FROM DISK FILE TO NA4 ARRAY.
MPK4 = NA4(NXA4)
J = 1
J = J+1
IS J > NXA4
Y
N
II (266Ø)
JJ (267Ø)
19

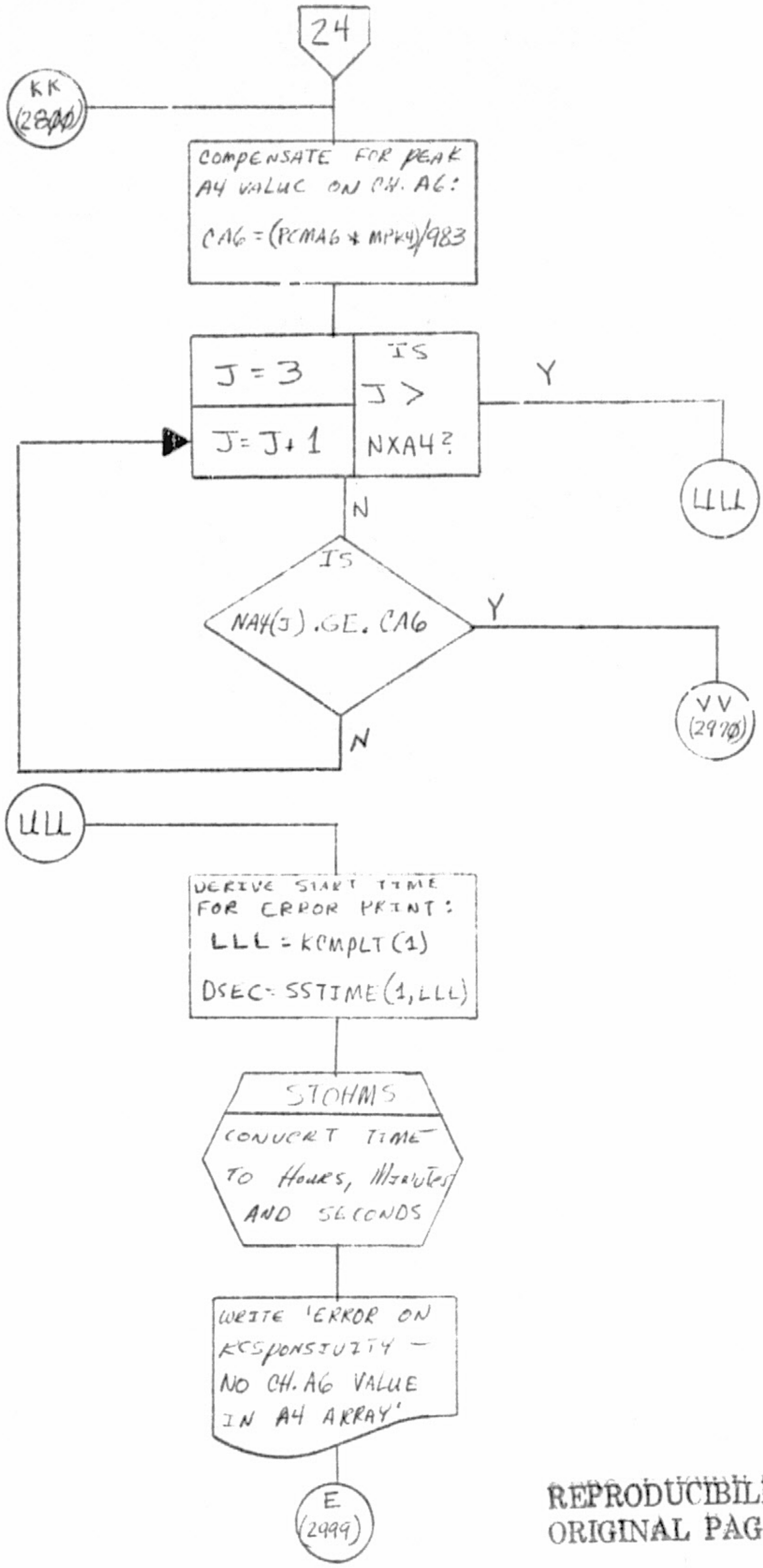

19
IS MPK4 .GT. NA4(K-J) ?
Y
N
JJ 267Ø
MPK4=NA4 (K-J)
II (266Ø)
JJ (267Ø)
CONTINUE
PROCESSING HEATED CAL DATA ?
Y
N
KK (28ØØ)
ADJUST FOR PEAK A4 OF 983 COUNTS:
CA2= (PCMA2*MPK4)/983
CA3= (PCMA3*MPK4)/983
CA5= (PCMA5*MPK4)/983
LL (271Ø)

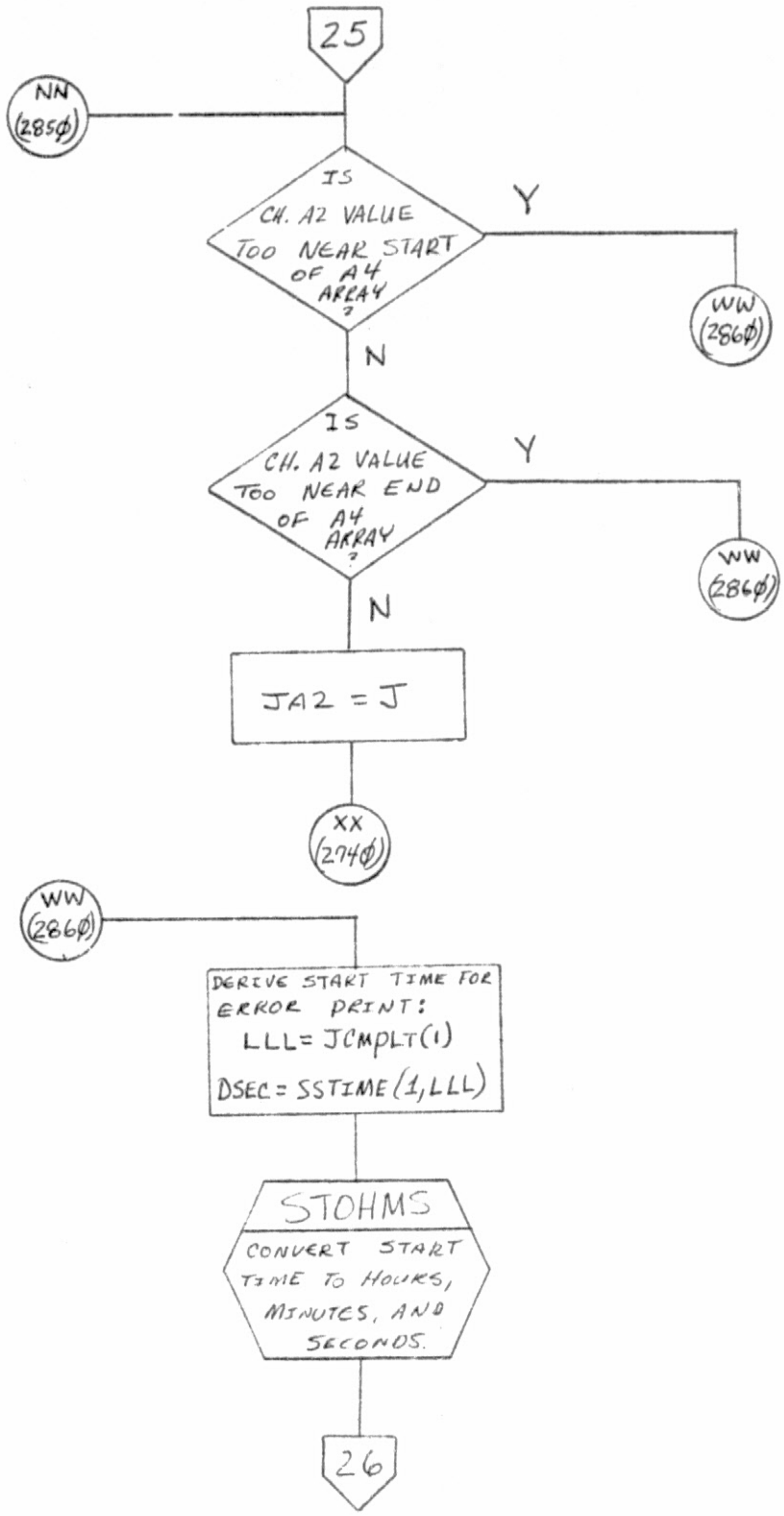

2Ø
P
(27ØØ)
ZERO
ZERO NA4
ARRAY FROM(1)
THRU (685)
WRITE ZEROS
IN CHANNEL
DATA ON
DISK FILE
IPSOK=Ø
NFSCN=Ø
NSAMPL=1
SET "PEAK A4 VALUE
NOT >956 AT END
OF SCAN" ANOMALY:
JPAR=18
KRET=1
NSAM=NX
LAN=2
SEG
CALLS
ANPRO.LLL
MODULE
MM
(214Ø)

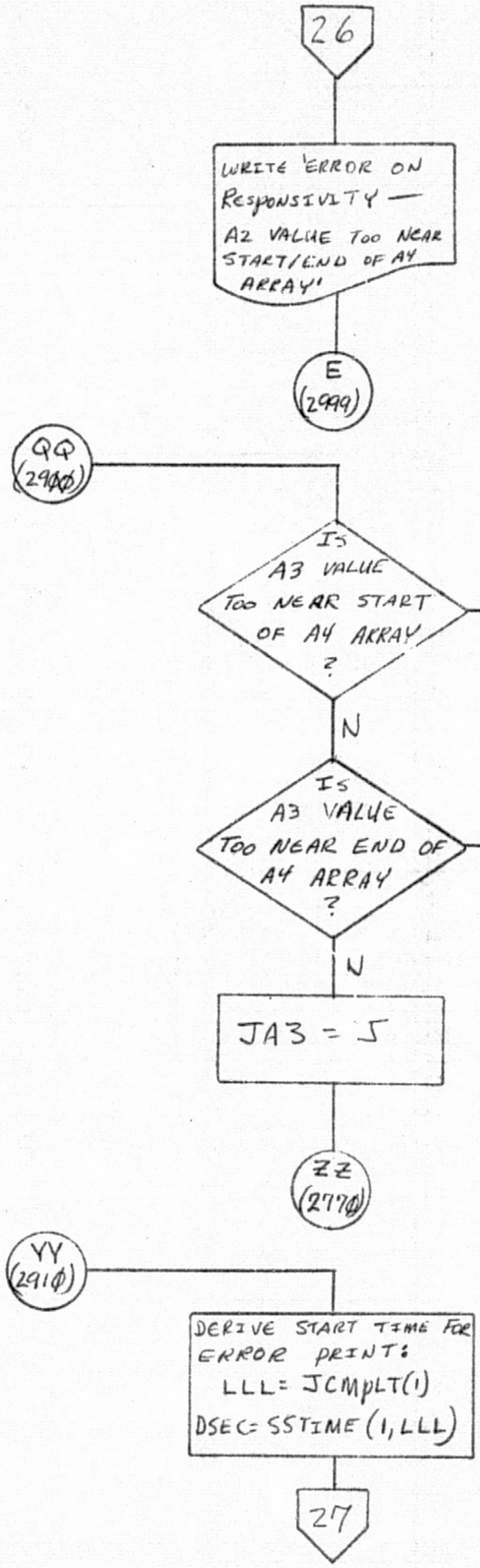

21
LL (271Ø)
J=3
J=J+1
IS J > NXA4 ?
Y
00
N
IS NA4(J) .GE. CA2 ?
Y
NN (285Ø)
N
00
Derive Time For ERROR PRINT:
LLL= JCMPLT(1)
DSEC= SSTIME(1,LLL)
STOHMS
CONVERT TIME To Hours, Mins, SECONDS
WRITE 'ERROR ON RESPONSIVITY - NO CH. A2 VALUE FOUND IN A4 ARRAY'
22


22
XX (274Ø)
J=3
J=J+1
IS J > NXA4?
Y
PP
N
IS NA4(J).GE. CA3
Y
QQ (290Ø)
N
PP
DERIVE START TIME TO PRINT ERROR:
LLL=JCMPLT(1)
DSEC=SSTIME(1,LLL)
WRITE 'ERROR DURING Responsivity NO CH. A3 VALUE IN A4 ARRAY
ZZ (277Ø)
J=3
J=J+1
IS J > NXA4?
KR
Y
SS
N
23

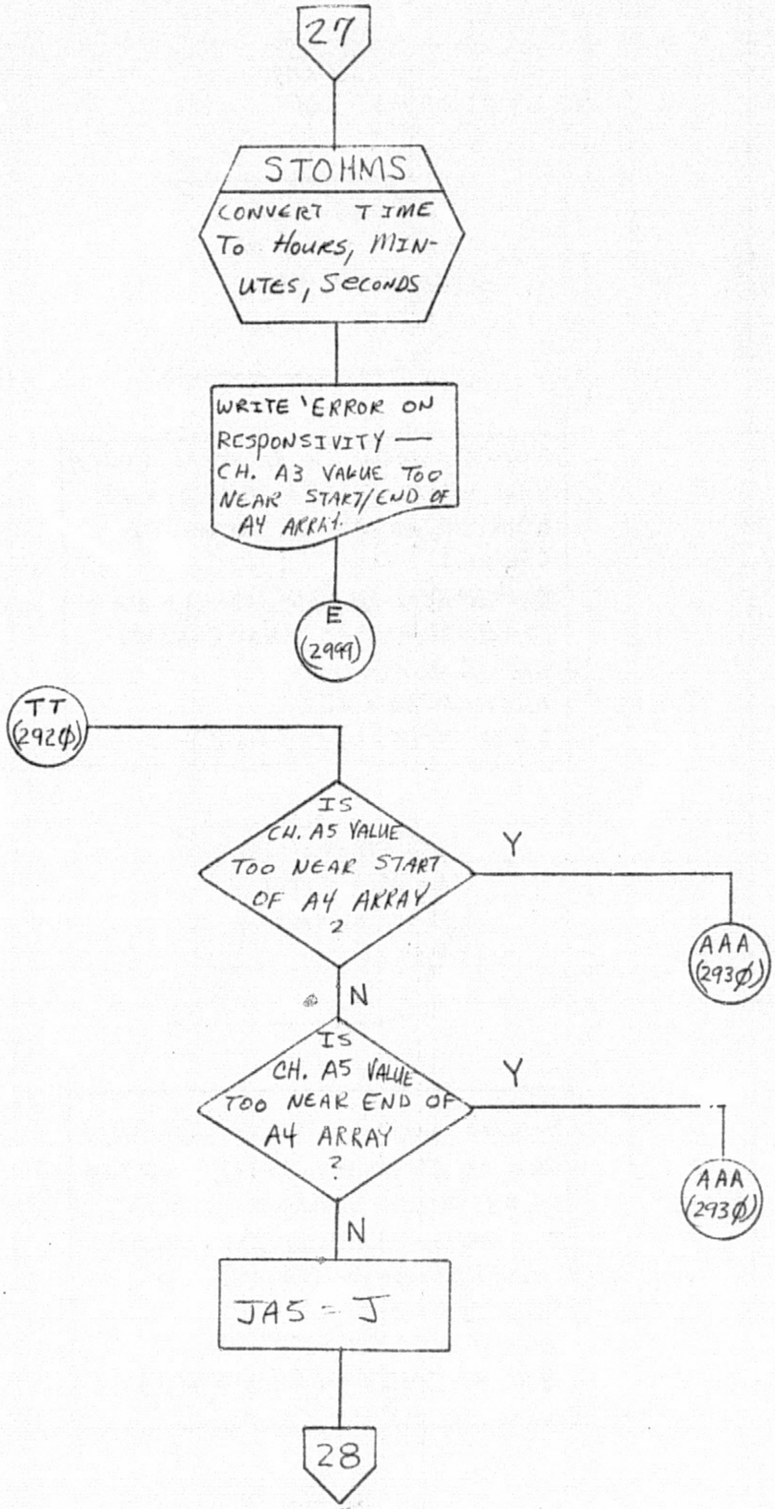

23
IS
NA4(J).GE. CA5
Y
N
TT
(292Ø)
RR
SS
DERIVE START TIME
TO PRINT ERROR:
LLL= JCMPLT(1)
DSEC=SSTIME(1,LLL)
STOHMS
CONVERT TIME TO
HOURS, MINUTES,
SECONDS
WRITE 'ERROR ON
RESPONSIVITY - NO
A5 VALUE IN A4
ARRAY'
E
(2999)

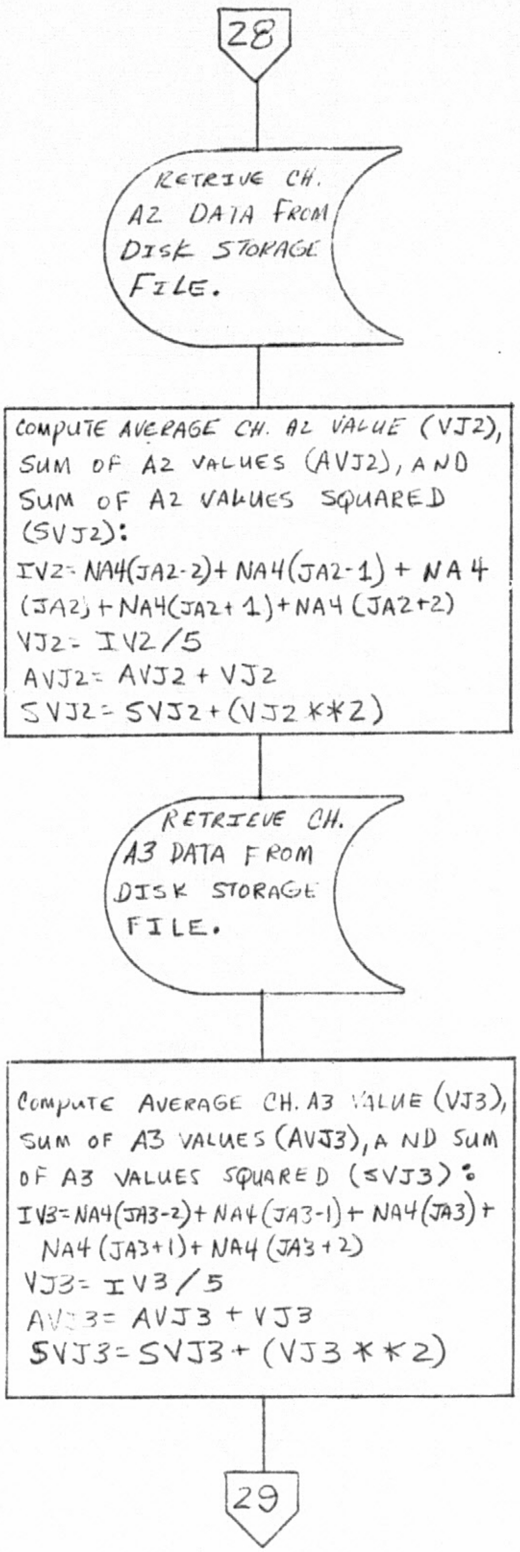

24
KK (28ØØ)
COMPENSATE FOR PEAK A4 VALUE ON CH. A6:
CA6 = (PCMA6 * MPK4)/983
J = 3
J = J + 1
IS J > NXA4?
Y
N
LLL
IS NA4(J) .GE. CA6
Y
N
VV (29ØØ)
LLL
DERIVE START TIME FOR ERROR PRINT:
LLL = KCMPLT (1)
DSEC = SSTIME (1, LLL)
STOHMS
CONVERT TIME TO HOURS, MINUTES AND SECONDS
WRITE 'ERROR ON RESPONSIBILITY — NO CH. A6 VALUE IN A4 ARRAY'
E (2999)

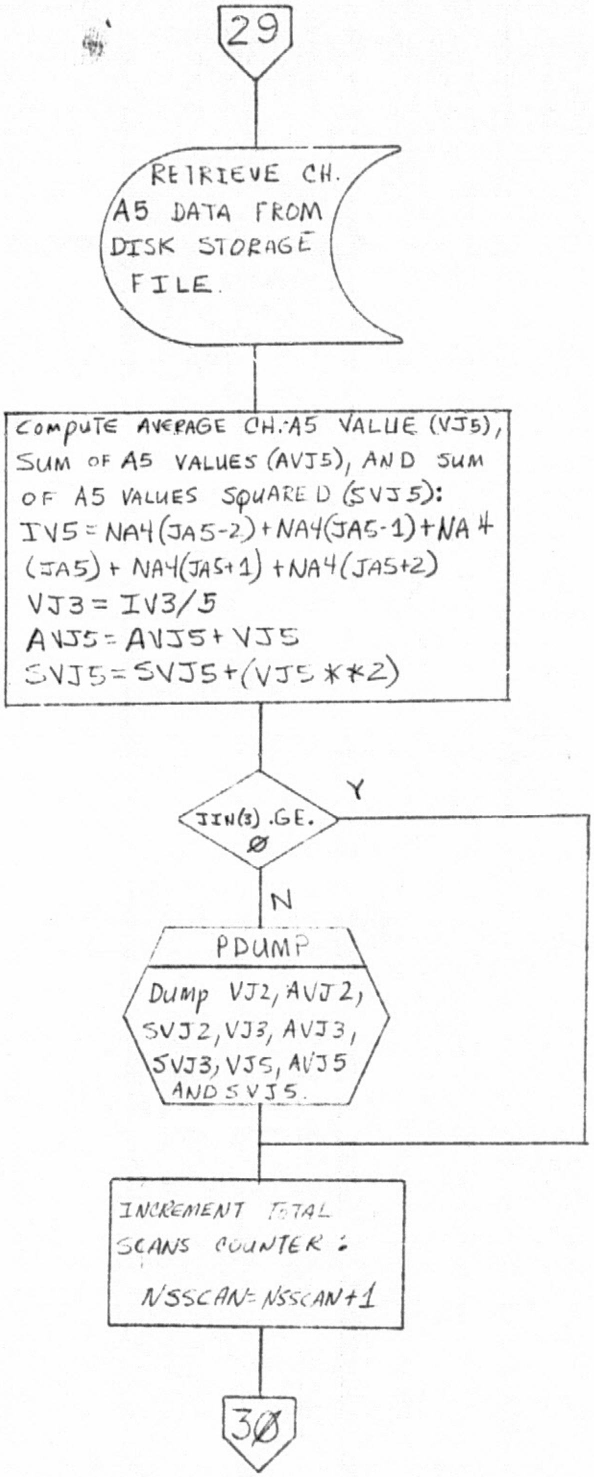

25
NN (285Ø)
IS CH. A2 VALUE TOO NEAR START OF A4 ARRAY ?
Y
WW (286Ø)
N
IS CH. A2 VALUE TOO NEAR END OF A4 ARRAY ?
Y
WW (286Ø)
N
JA2 = J
XX (274Ø)
WW (286Ø)
DERIVE START TIME FOR ERROR PRINT:
LLL= JCMPLT(1)
DSEC = SSTIME (1,LLL)
STOHMS
CONVERT START TIME TO HOURS, MINUTES, AND SECONDS.
26

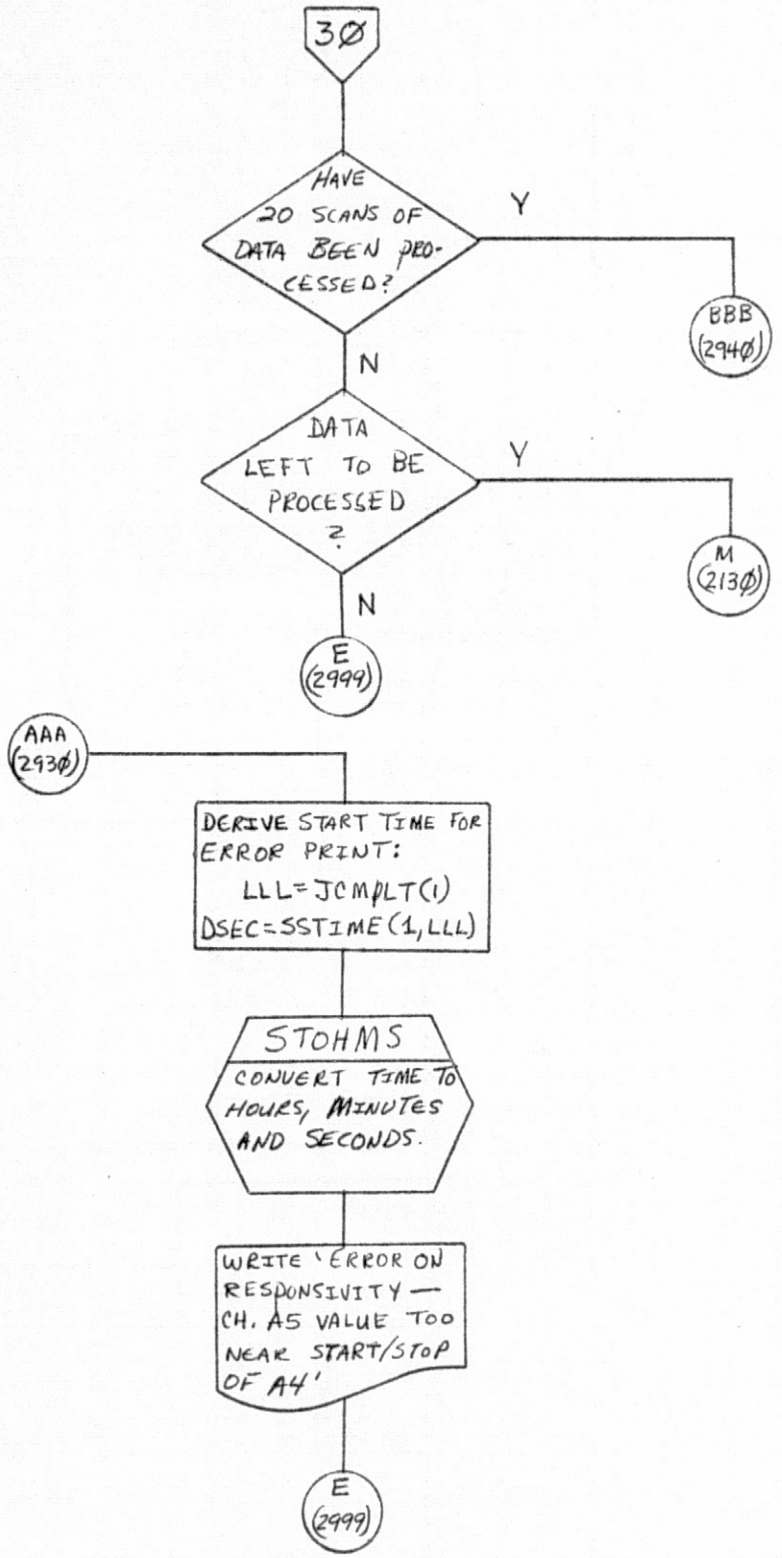

26
WRITE 'ERROR ON RESPONSIVITY — A2 VALUE TOO NEAR START/END OF A4 ARRAY'
E (2999)
QQ (2900)
IS A3 VALUE TOO NEAR START OF A4 ARRAY ?
Y
YY (2910)
N
IS A3 VALUE TOO NEAR END OF A4 ARRAY ?
Y
YY (2910)
N
JA3 = J
ZZ (2770)
YY (2910)
DERIVE START TIME FOR ERROR PRINT:
LLL= JCMPLT(1)
DSEC= SSTIME (1,LLL)
27

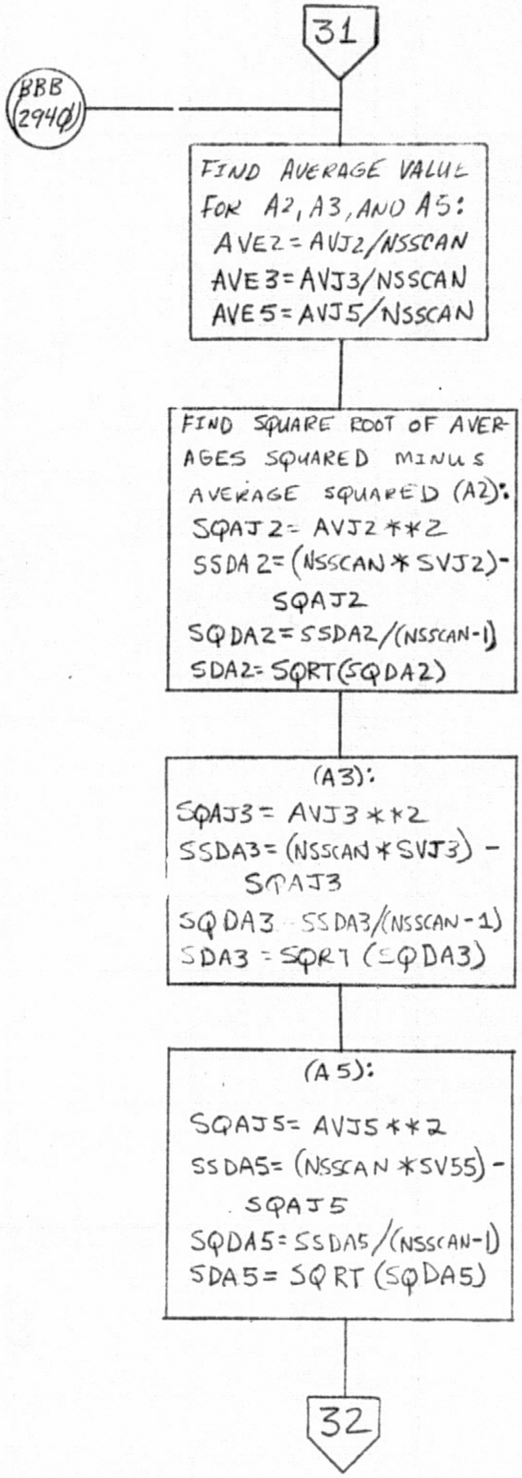
C-4

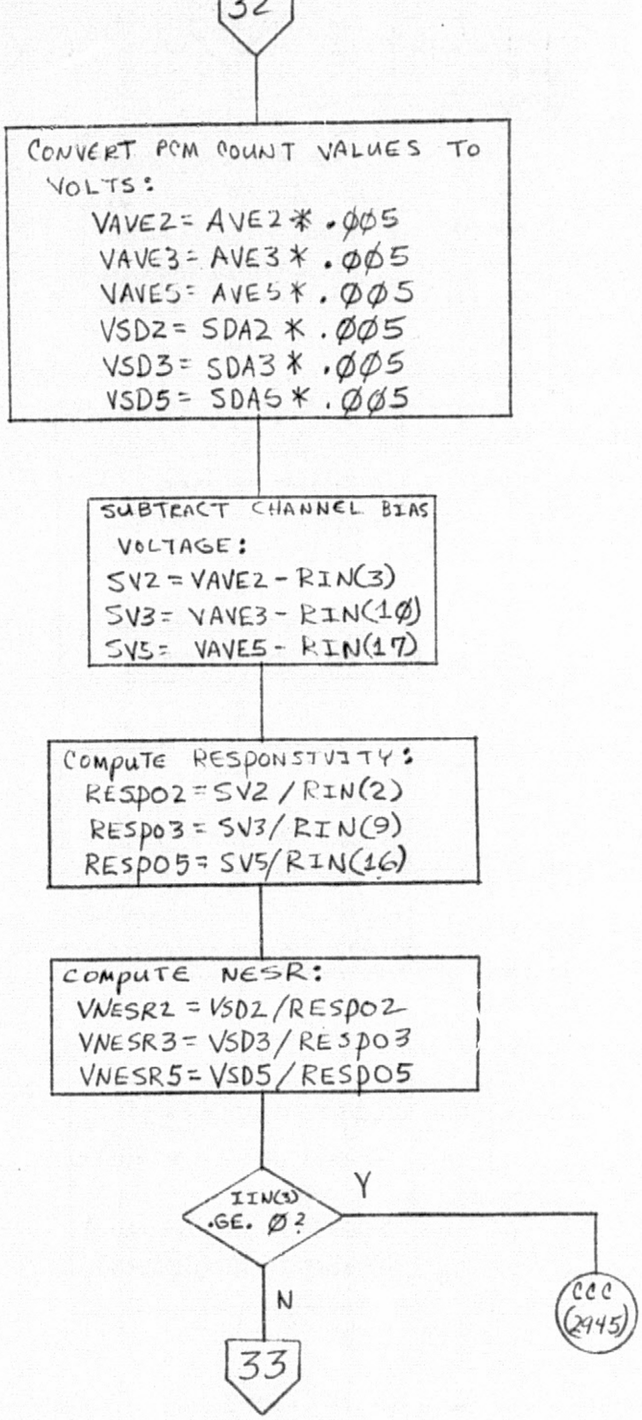

27
STOHMS
CONVERT TIME To Hours, MIN-UTES, SECONDS
WRITE 'ERROR ON RESPONSIVITY — CH. A3 VALUE TOO NEAR START/END OF A4 ARRAY'
E (2999)
TT (292Ø)
IS CH. A5 VALUE TOO NEAR START OF A4 ARRAY ?
Y
AAA (293Ø)
N
IS CH. A5 VALUE TOO NEAR END OF A4 ARRAY ?
Y
AAA (293Ø)
N
JA5 = J
28

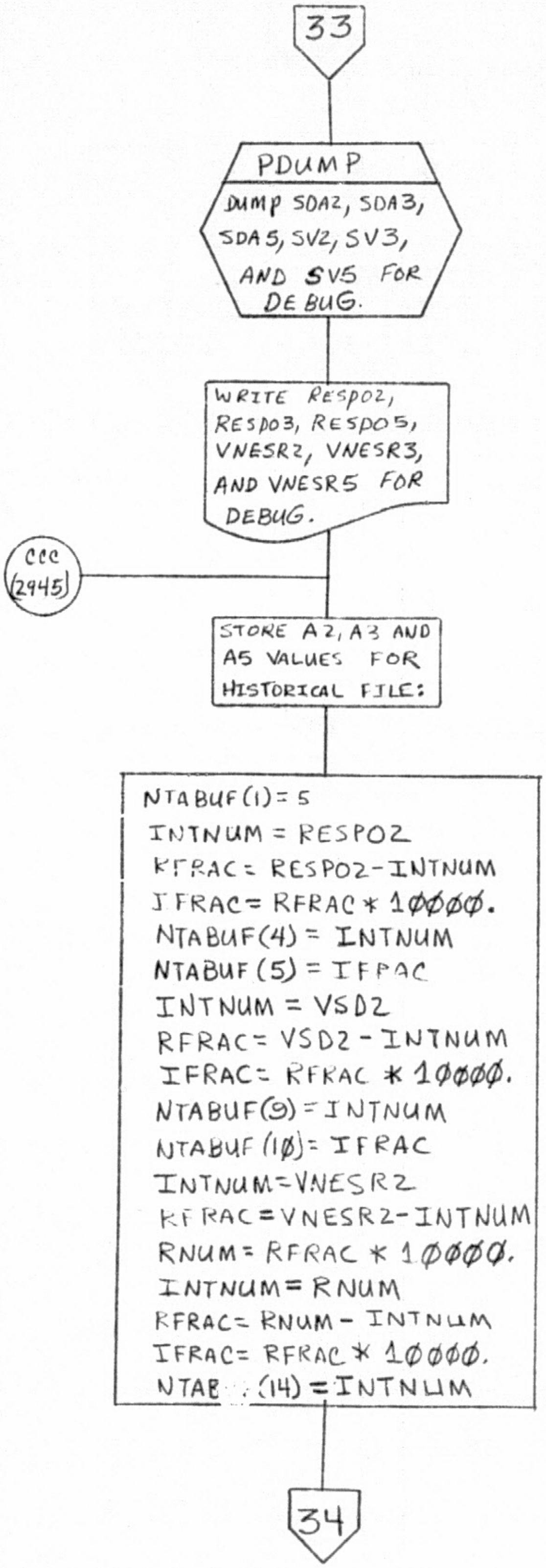

28
RETRIEVE CH. A2 DATA FROM DISK STORAGE FILE.
COMPUTE AVERAGE CH. A2 VALUE (VJ2), SUM OF A2 VALUES (AVJ2), AND SUM OF A2 VALUES SQUARED (SVJ2):
IV2 = NA4(JA2-2) + NA4(JA2-1) + NA4 (JA2) + NA4(JA2+1) + NA4 (JA2+2)
VJ2 = IV2/5
AVJ2 = AVJ2 + VJ2
SVJ2 = SVJ2 + (VJ2**2)
RETRIEVE CH. A3 DATA FROM DISK STORAGE FILE.
COMPUTE AVERAGE CH. A3 VALUE (VJ3), SUM OF A3 VALUES (AVJ3), AND SUM OF A3 VALUES SQUARED (SVJ3):
IV3 = NA4(JA3-2) + NA4(JA3-1) + NA4(JA3) + NA4 (JA3+1) + NA4 (JA3+2)
VJ3 = IV3/5
AVJ3 = AVJ3 + VJ3
SVJ3 = SVJ3 + (VJ3**2)
29

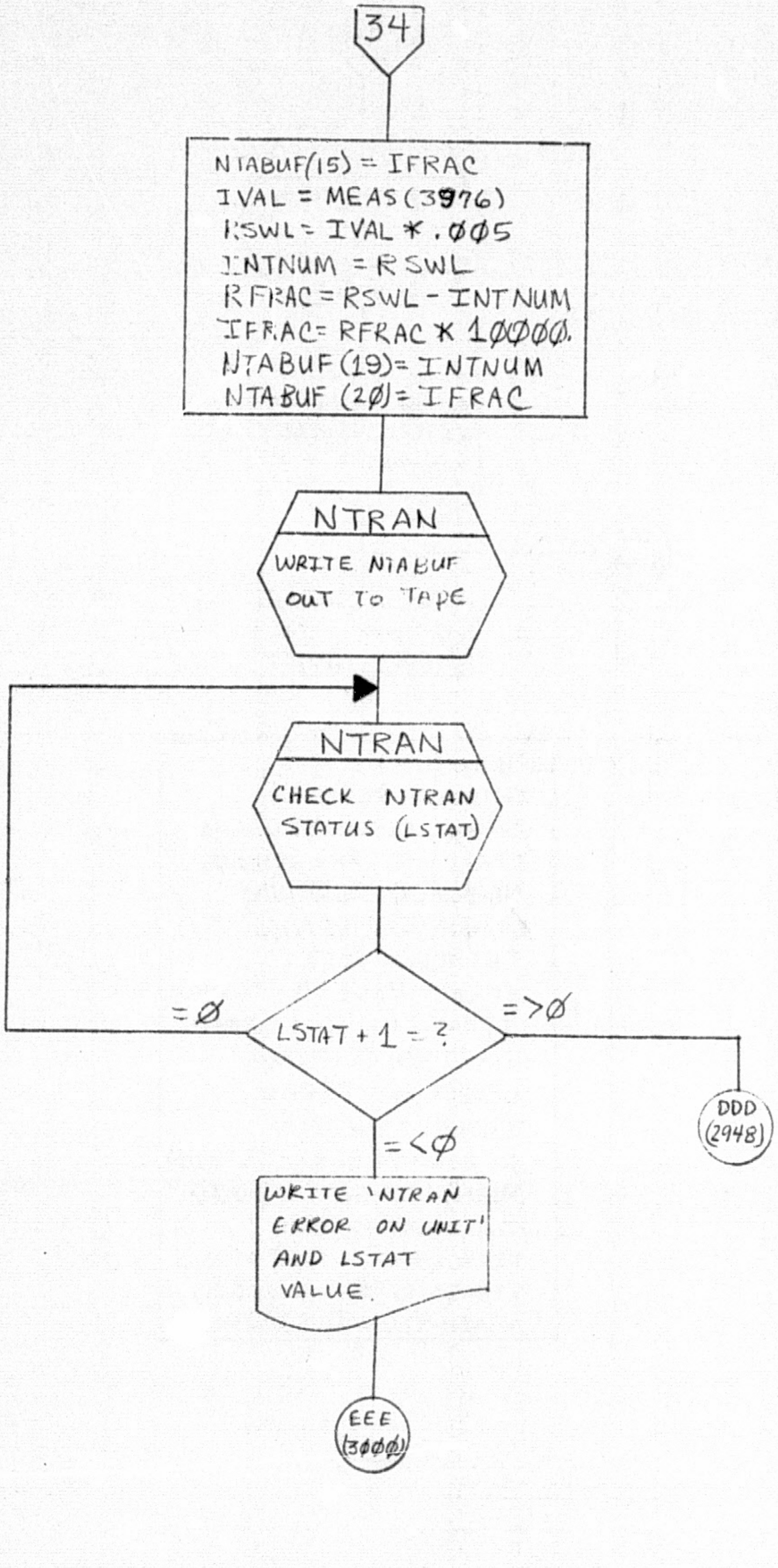

29
RETRIEVE CH. A5 DATA FROM DISK STORAGE FILE.
COMPUTE AVERAGE CH. A5 VALUE (VJ5), SUM OF A5 VALUES (AVJ5), AND SUM OF A5 VALUES SQUARED (SVJ5):
IV5 = NA4(JA5-2) + NA4(JA5-1) + NA4 (JA5) + NA4(JA5+1) + NA4(JA5+2)
VJ3 = IV3/5
AVJ5 = AVJ5 + VJ5
SVJ5 = SVJ5 + (VJ5 ** 2)
JIN(3) .GE. Ø
Y
N
PDUMP
Dump VJ2, AVJ2, SVJ2, VJ3, AVJ3, SVJ3, VJ5, AVJ5 AND SVJ5.
INCREMENT TOTAL SCANS COUNTER :
NSSCAN = NSSCAN + 1
3Ø

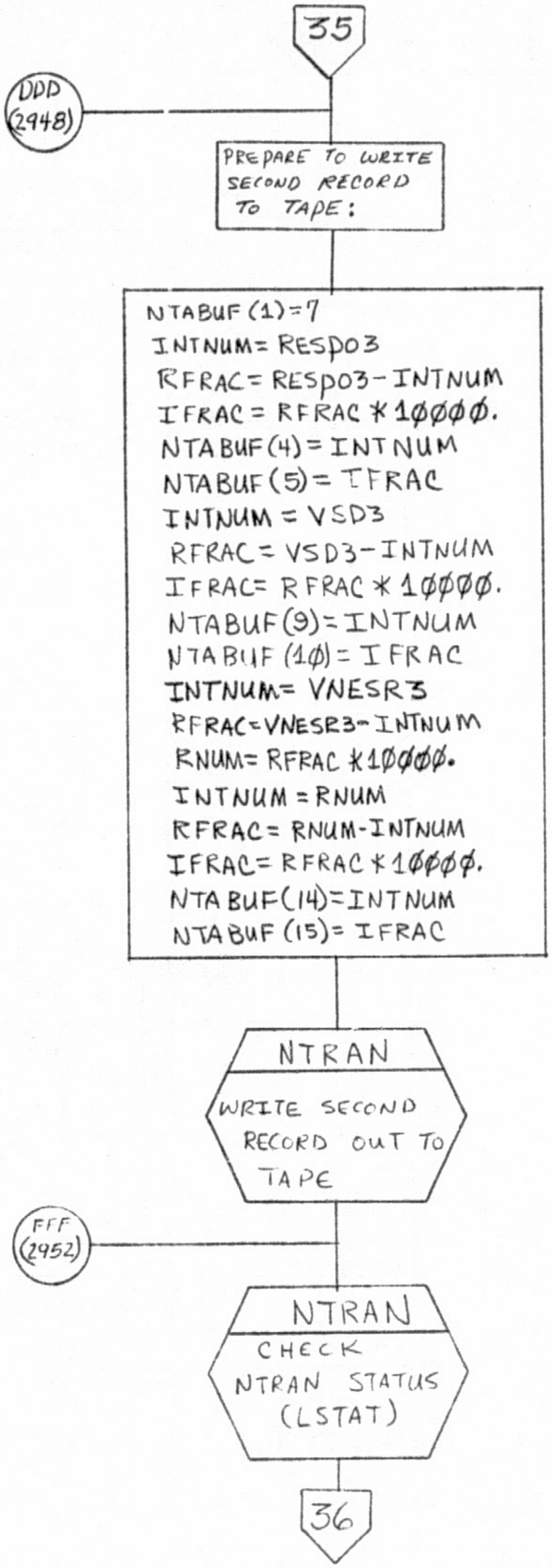

30
HAVE 20 SCANS OF DATA BEEN PROCESSED?
Y
BBB (2940)
N
DATA LEFT TO BE PROCESSED ?
Y
M (2130)
N
E (2999)
AAA (2930)
DERIVE START TIME FOR ERROR PRINT:
LLL=JCMPLT(1)
DSEC=SSTIME(1,LLL)
STOHMS
CONVERT TIME TO HOURS, MINUTES AND SECONDS.
WRITE 'ERROR ON RESPONSIVITY — CH. A5 VALUE TOO NEAR START/STOP OF A4'
E (2999)

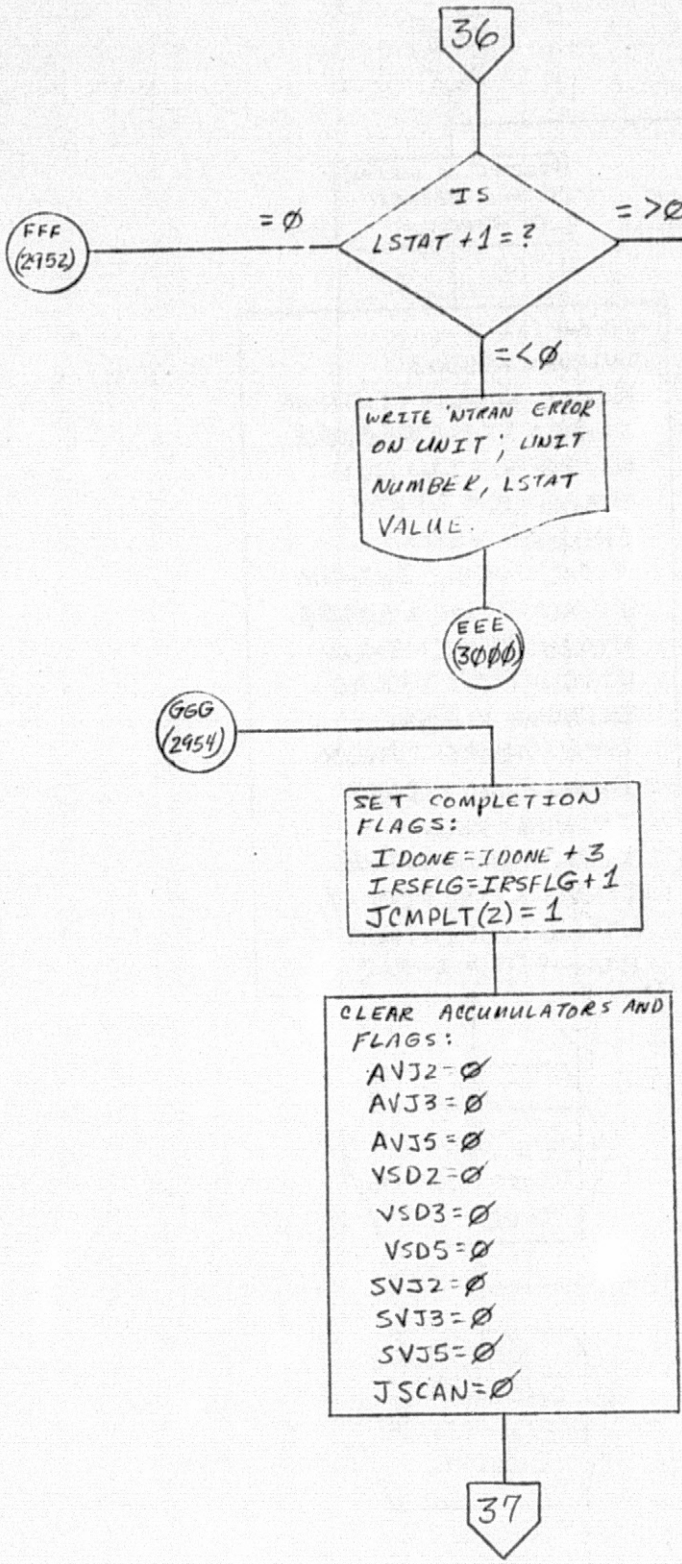

31
BBB (294Ø)
FIND AVERAGE VALUE FOR A2, A3, AND A5:
AVE2 = AVJ2/NSSCAN
AVE3 = AVJ3/NSSCAN
AVE5 = AVJ5/NSSCAN
FIND SQUARE ROOT OF AVERAGES SQUARED MINUS AVERAGE SQUARED (A2):
SQAJ2 = AVJ2 **2
SSDA2 = (NSSCAN * SVJ2) - SQAJ2
SQDA2 = SSDA2/(NSSCAN-1)
SDA2 = SQRT(SQDA2)
(A3):
SQAJ3 = AVJ3 **2
SSDA3 = (NSSCAN * SVJ3) - SQAJ3
SQDA3 = SSDA3/(NSSCAN - 1)
SDA3 = SQRT (SQDA3)
(A5):
SQAJ5 = AVJ5 **2
SSDA5 = (NSSCAN *SV55) - SQAJ5
SQDA5 = SSDA5/(NSSCAN-1)
SDA5 = SQRT (SQDA5)
32

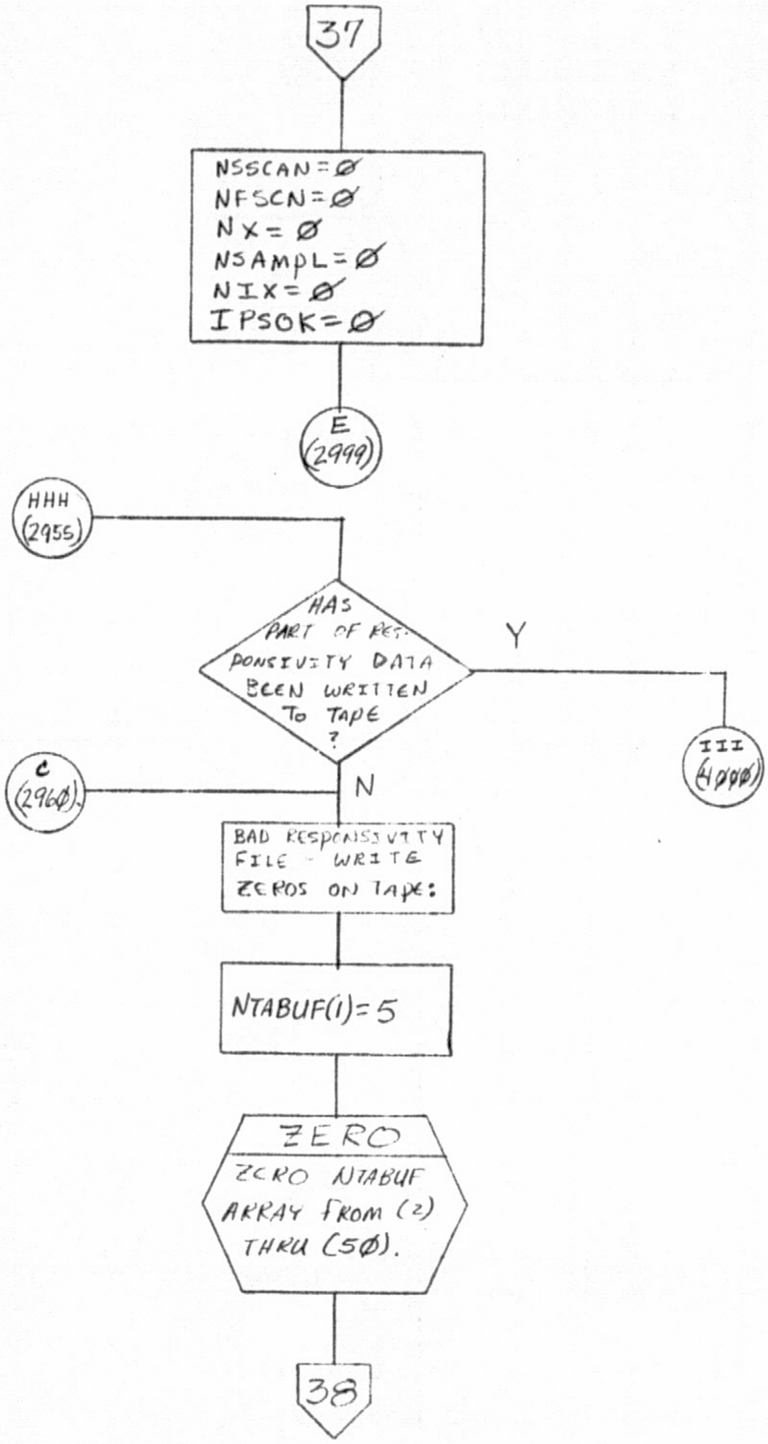

32
CONVERT PCM COUNT VALUES TO VOLTS:
VAVE2 = AVE2 * .ØØ5
VAVE3 = AVE3 * .ØØ5
VAVE5 = AVE5 * .ØØ5
VSD2 = SDA2 * .ØØ5
VSD3 = SDA3 * .ØØ5
VSD5 = SDA5 * .ØØ5
SUBTRACT CHANNEL BIAS VOLTAGE:
SV2 = VAVE2 - RIN(3)
SV3 = VAVE3 - RIN(1Ø)
SV5 = VAVE5 - RIN(17)
COMPUTE RESPONSIVITY:
RESPO2 = SV2 / RIN(2)
RESPO3 = SV3 / RIN(9)
RESPO5 = SV5 / RIN(16)
COMPUTE NESR:
VNESR2 = VSD2 / RESPO2
VNESR3 = VSD3 / RESPO3
VNESR5 = VSD5 / RESPO5
IIN(3) .GE. Ø?
Y
N
CCC (2945)
33

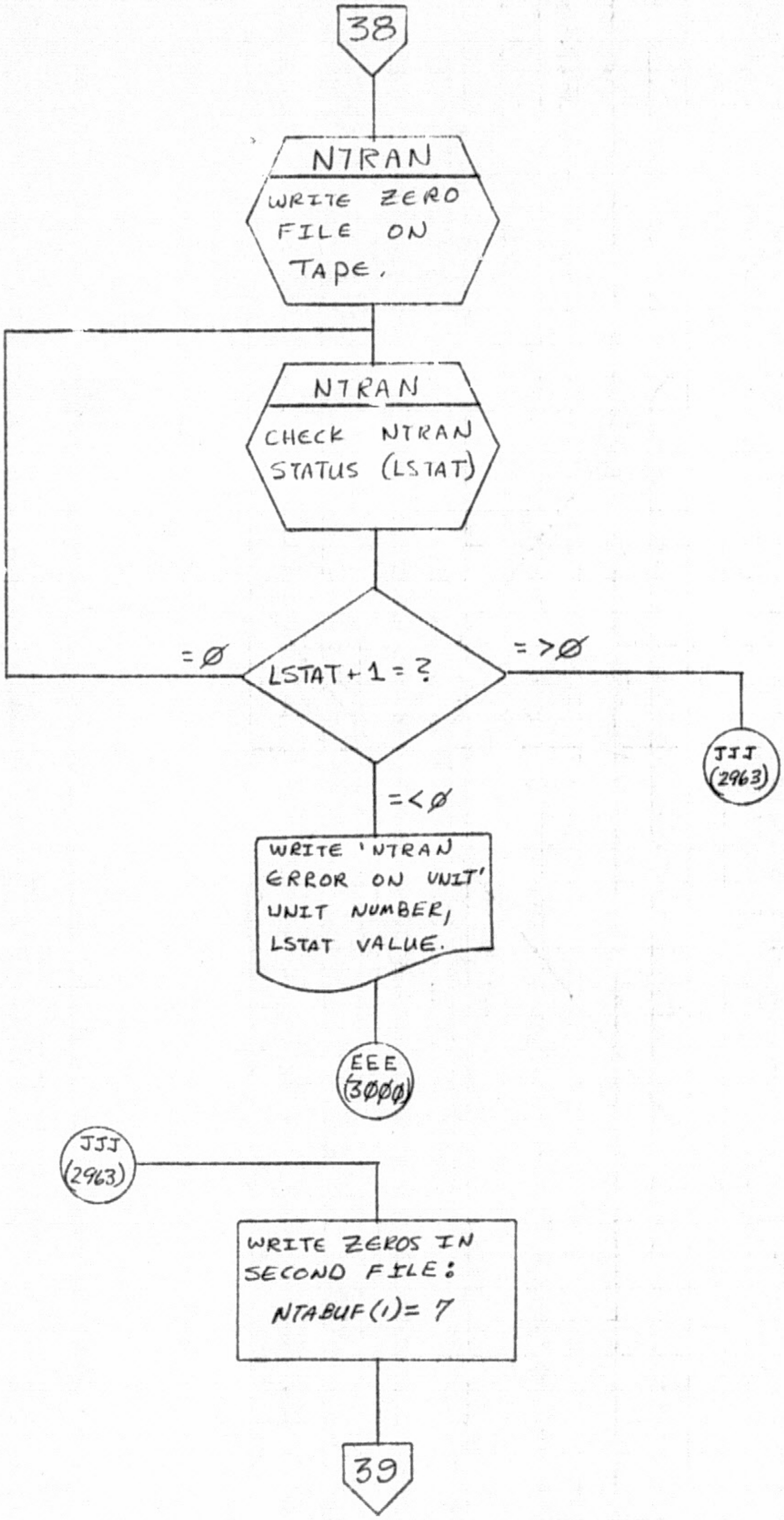

33
PDUMP
DUMP SDA2, SDA3, SDA5, SV2, SV3, AND SV5 FOR DEBUG.
WRITE RESPO2, RESPO3, RESPO5, VNESR2, VNESR3, AND VNESR5 FOR DEBUG.
CCC (2945)
STORE A2, A3 AND A5 VALUES FOR HISTORICAL FILE:
NTABUF(1)=5
INTNUM = RESPO2
RFRAC= RESPO2-INTNUM
IFRAC= RFRAC * 10000.
NTABUF(4) = INTNUM
NTABUF(5) = IFRAC
INTNUM = VSD2
RFRAC= VSD2 - INTNUM
IFRAC= RFRAC * 10000.
NTABUF(9) = INTNUM
NTABUF(10)= IFRAC
INTNUM=VNESR2
RFRAC=VNESR2-INTNUM
RNUM= RFRAC * 10000.
INTNUM=RNUM
RFRAC= RNUM - INTNUM
IFRAC= RFRAC * 10000.
NTAB (14) =INTNUM
34

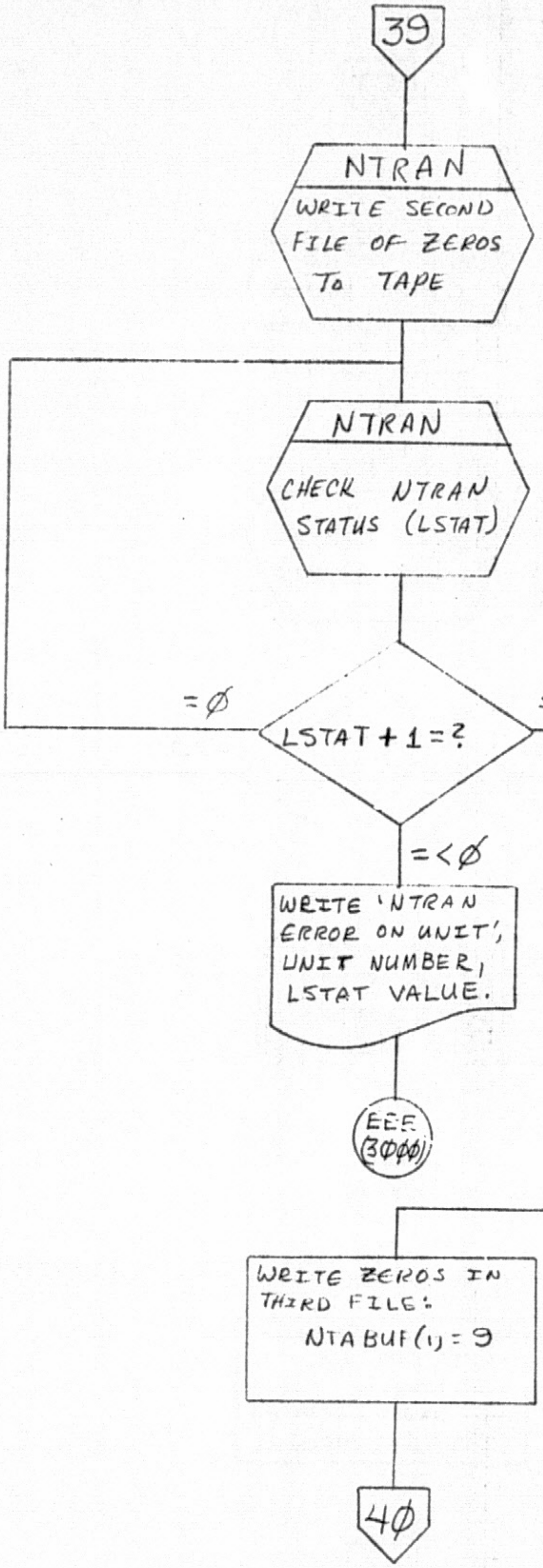

34
NTABUF(15) = IFRAC
IVAL = MEAS(3976)
RSWL = IVAL * .ØØ5
INTNUM = RSWL
RFRAC = RSWL - INTNUM
IFRAC = RFRAC * 1ØØØØ.
NTABUF(19) = INTNUM
NTABUF(2Ø) = IFRAC
NTRAN
WRITE NTABUF OUT TO TAPE
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT + 1 = ?
= Ø
= > Ø
= < Ø
DDD (2948)
WRITE 'NTRAN ERROR ON UNIT' AND LSTAT VALUE.
EEE (3ØØØ)

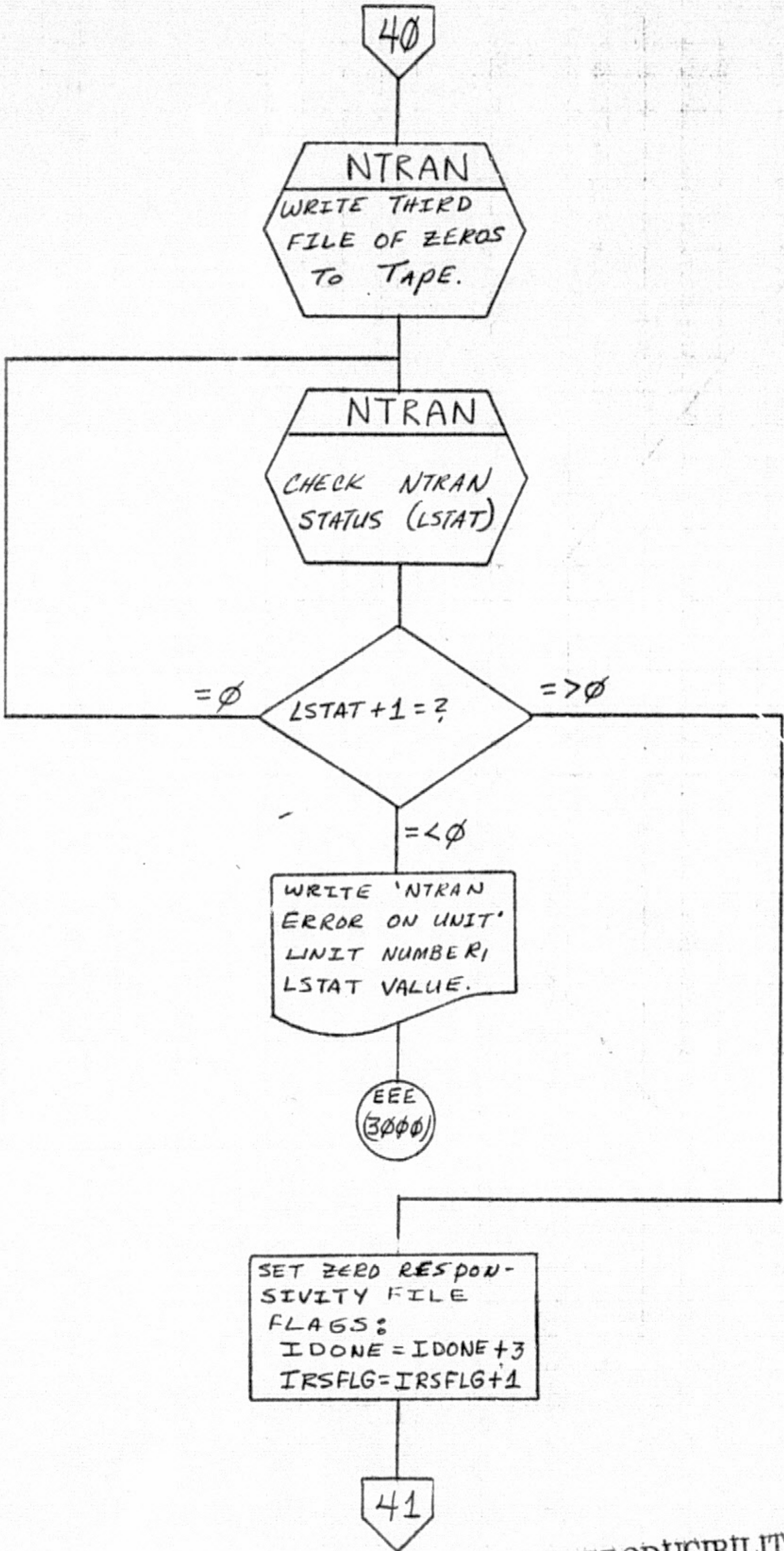

35
DDD (2948)
PREPARE TO WRITE SECOND RECORD TO TAPE:
NTABUF(1)=7
INTNUM=RESPO3
RFRAC=RESPO3-INTNUM
IFRAC=RFRAC*1ØØØØ.
NTABUF(4)=INTNUM
NTABUF(5)=IFRAC
INTNUM=VSD3
RFRAC=VSD3-INTNUM
IFRAC=RFRAC*1ØØØØ.
NTABUF(9)=INTNUM
NTABUF(1Ø)=IFRAC
INTNUM=VNESR3
RFRAC=VNESR3-INTNUM
RNUM=RFRAC*1ØØØØ.
INTNUM=RNUM
RFRAC=RNUM-INTNUM
IFRAC=RFRAC*1ØØØØ.
NTABUF(14)=INTNUM
NTABUF(15)=IFRAC
NTRAN
WRITE SECOND RECORD OUT TO TAPE
FFF (2952)
NTRAN
CHECK NTRAN STATUS (LSTAT)
36

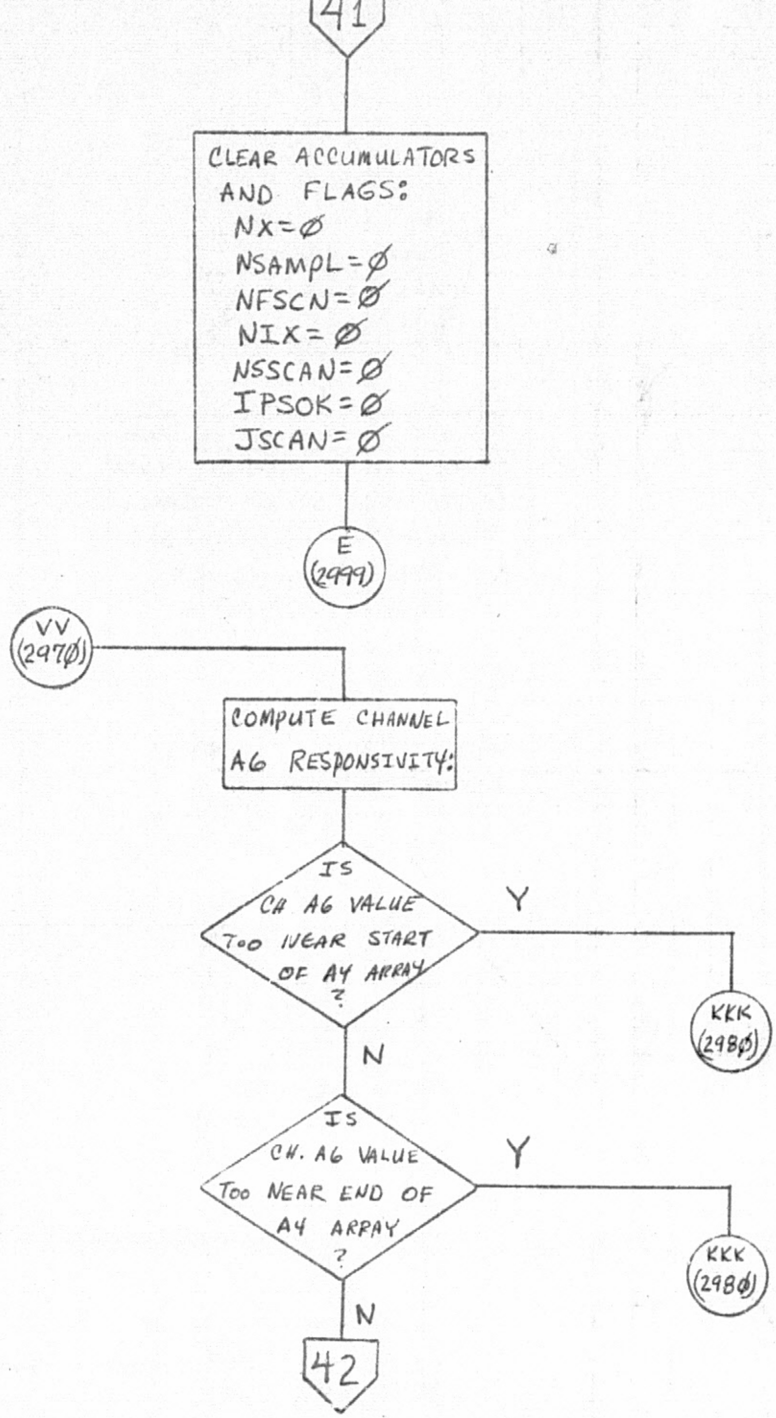

36
IS LSTAT + 1 = ?
= Ø
= >Ø
= <Ø
FFF (2952)
GGG (2954)
WRITE 'NTRAN ERROR ON UNIT'; UNIT NUMBER, LSTAT VALUE.
EEE (3ØØØ)
GGG (2954)
SET COMPLETION FLAGS:
IDONE = IDONE + 3
IRSFLG = IRSFLG + 1
JCMPLT(2) = 1
CLEAR ACCUMULATORS AND FLAGS:
AVJ2 = Ø
AVJ3 = Ø
AVJ5 = Ø
VSD2 = Ø
VSD3 = Ø
VSD5 = Ø
SVJ2 = Ø
SVJ3 = Ø
SVJ5 = Ø
JSCAN = Ø
37

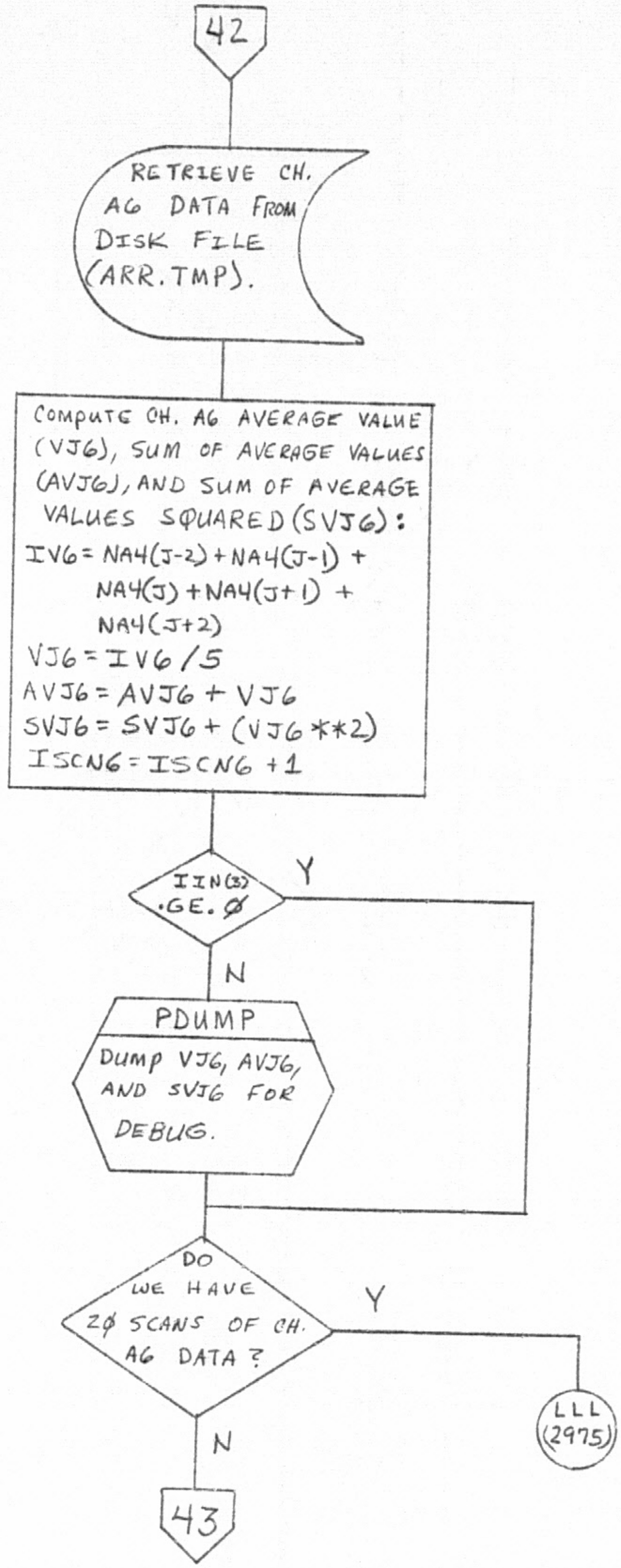

37
NSSCAN=Ø
NFSCN=Ø
NX=Ø
NSAMPL=Ø
NIX=Ø
IPSOK=Ø
E (2999)
HHH (2955)
HAS PART OF RESPONSIVITY DATA BEEN WRITTEN TO TAPE ?
Y
III (4000)
N
C (2960)
BAD RESPONSIVITY FILE - WRITE ZEROS ON TAPE:
NTABUF(1)=5
ZERO
ZERO NTABUF ARRAY FROM (2) THRU (50).
38

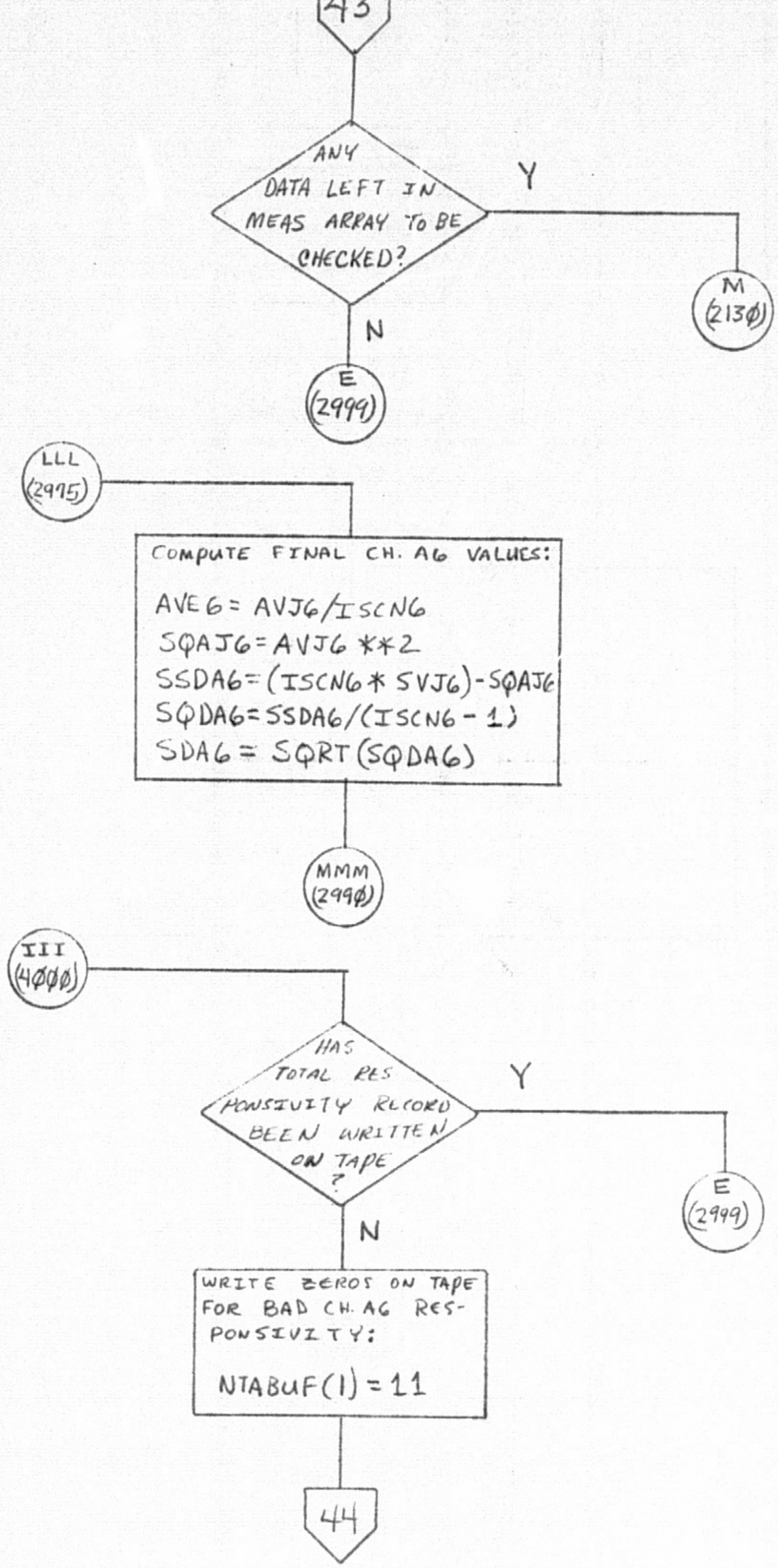

38
NTRAN
WRITE ZERO FILE ON TAPE.
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT + 1 = ?
= Ø
= > Ø
= < Ø
JJJ (2963)
WRITE 'NTRAN ERROR ON UNIT' UNIT NUMBER, LSTAT VALUE.
EEE (3ØØØ)
JJJ (2963)
WRITE ZEROS IN SECOND FILE:
NTABUF(1)= 7
39

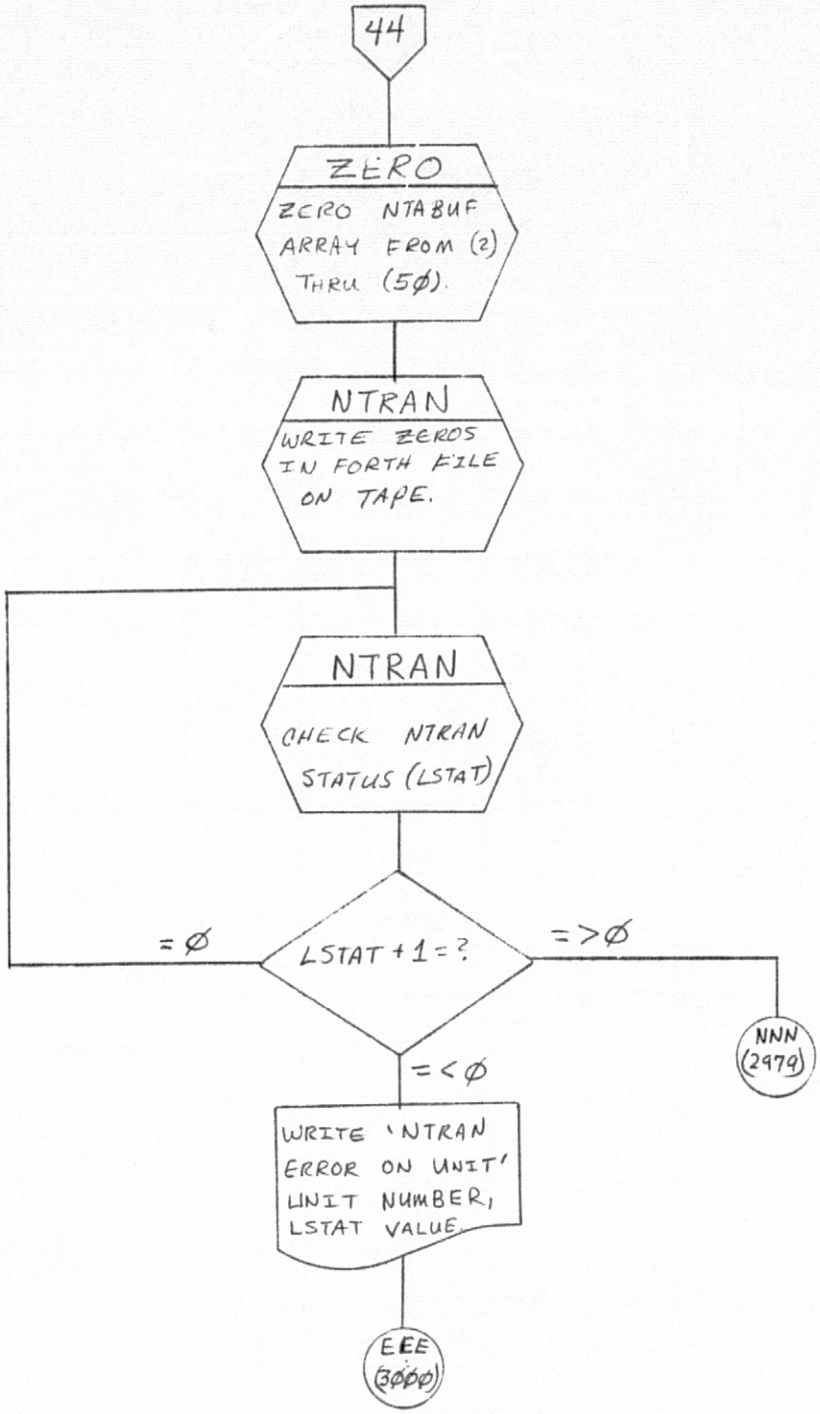

39
NTRAN
WRITE SECOND FILE OF ZEROS TO TAPE
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT + 1 = ?
= Ø
=> Ø
=< Ø
WRITE 'NTRAN ERROR ON UNIT', UNIT NUMBER, LSTAT VALUE.
EEE (3ØØØ)
WRITE ZEROS IN THIRD FILE: NTABUF(1) = 9
4Ø

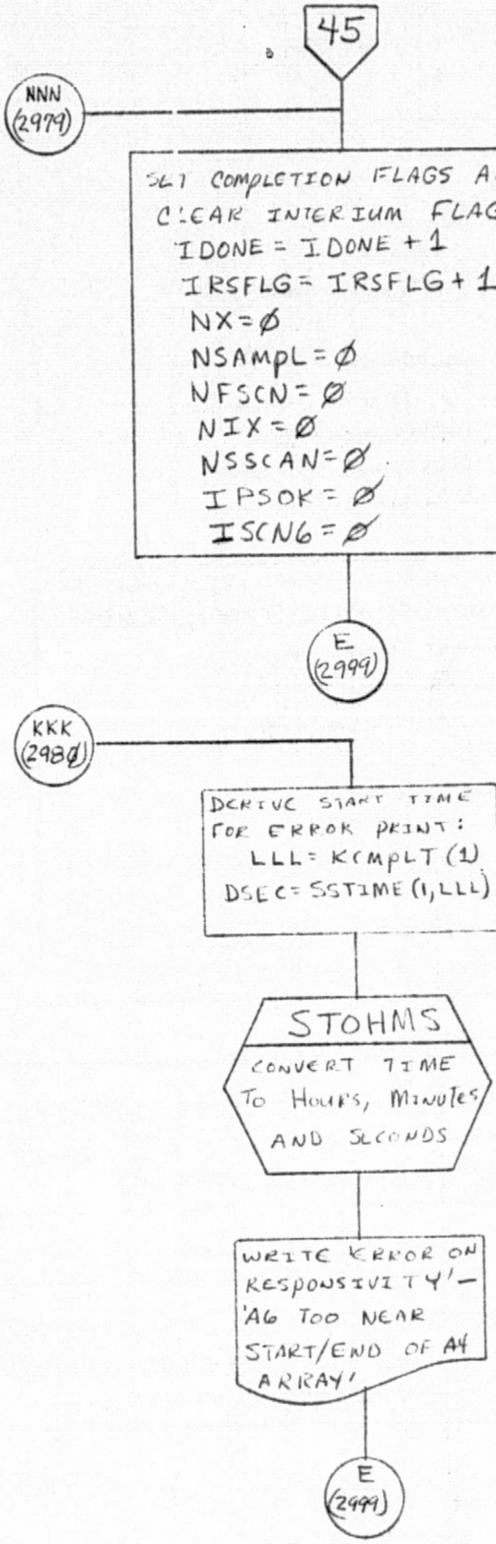

40
NTRAN
WRITE THIRD FILE OF ZEROS TO TAPE.
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT + 1 = ?
= Ø
=> Ø
=< Ø
WRITE 'NTRAN ERROR ON UNIT' UNIT NUMBER, LSTAT VALUE.
EEE (3ØØØ)
SET ZERO RESPONSIVITY FILE FLAGS:
IDONE = IDONE + 3
IRSFLG = IRSFLG + 1
41


41
CLEAR ACCUMULATORS AND FLAGS:
NX=Ø
NSAMPL=Ø
NFSCN=Ø
NIX=Ø
NSSCAN=Ø
IPSOK=Ø
JSCAN=Ø
E (2999)
VV (297Ø)
COMPUTE CHANNEL A6 RESPONSIVITY:
IS CH A6 VALUE TOO NEAR START OF A4 ARRAY ?
Y
KKK (298Ø)
N
IS CH. A6 VALUE TOO NEAR END OF A4 ARRAY ?
Y
KKK (298Ø)
N
42


42
RETRIEVE CH. A6 DATA FROM DISK FILE (ARR.TMP).
COMPUTE CH. A6 AVERAGE VALUE (VJ6), SUM OF AVERAGE VALUES (AVJ6), AND SUM OF AVERAGE VALUES SQUARED (SVJ6):
IV6 = NA4(J-2) + NA4(J-1) + NA4(J) + NA4(J+1) + NA4(J+2)
VJ6 = IV6/5
AVJ6 = AVJ6 + VJ6
SVJ6 = SVJ6 + (VJ6**2)
ISCN6 = ISCN6 + 1
IIN(3) .GE. Ø
Y
N
PDUMP
DUMP VJ6, AVJ6, AND SVJ6 FOR DEBUG.
DO WE HAVE 2Ø SCANS OF CH. A6 DATA?
Y
N
LLL (2975)
43


43
ANY DATA LEFT IN MEAS ARRAY TO BE CHECKED?
Y
M (213Ø)
N
E (2999)
LLL (2995)
COMPUTE FINAL CH. A6 VALUES:
AVE6 = AVJ6/ISCN6
SØAJ6 = AVJ6 **2
SSDA6 = (ISCN6 * SVJ6) - SØAJ6
SØDA6 = SSDA6/(ISCN6 - 1)
SDA6 = SQRT (SØDA6)
MMM (299Ø)
III (4ØØØ)
HAS TOTAL RESPONSIVITY RECORD BEEN WRITTEN ON TAPE ?
Y
E (2999)
N
WRITE ZEROS ON TAPE FOR BAD CH. A6 RESPONSIVITY:
NTABUF(1) = 11
44


44
ZERO
ZERO NTABUF ARRAY FROM (2) THRU (5Ø).
NTRAN
WRITE ZEROS IN FORTH FILE ON TAPE.
NTRAN
CHECK NTRAN STATUS (LSTAT)
LSTAT + 1 = ?
= Ø
=>Ø
=<Ø
NNN (2979)
WRITE 'NTRAN ERROR ON UNIT' UNIT NUMBER, LSTAT VALUE.
EEE (3ØØØ)


45
NNN (2979)
SET COMPLETION FLAGS AND CLEAR INTERIUM FLAGS:
IDONE = IDONE + 1
IRSFLG = IRSFLG + 1
NX = Ø
NSAMPL = Ø
NFSCN = Ø
NIX = Ø
NSSCAN = Ø
IPSOK = Ø
ISCN6 = Ø
E (2999)
KKK (298Ø)
DERIVE START TIME FOR ERROR PRINT:
LLL = KCMPLT (1)
DSEC = SSTIME (1,LLL)
STOHMS
CONVERT TIME TO HOURS, MINUTES AND SECONDS
WRITE 'ERROR ON RESPONSIVITY' — 'A6 TOO NEAR START/END OF A4 ARRAY'
E (2999)

46

MMM (299Ø)

CONVERT A6 PCM VALUE TO VOLTS:
VAVE6 = AVE6 * .ØØ5
VSD6 = SDA6 * .ØØ5

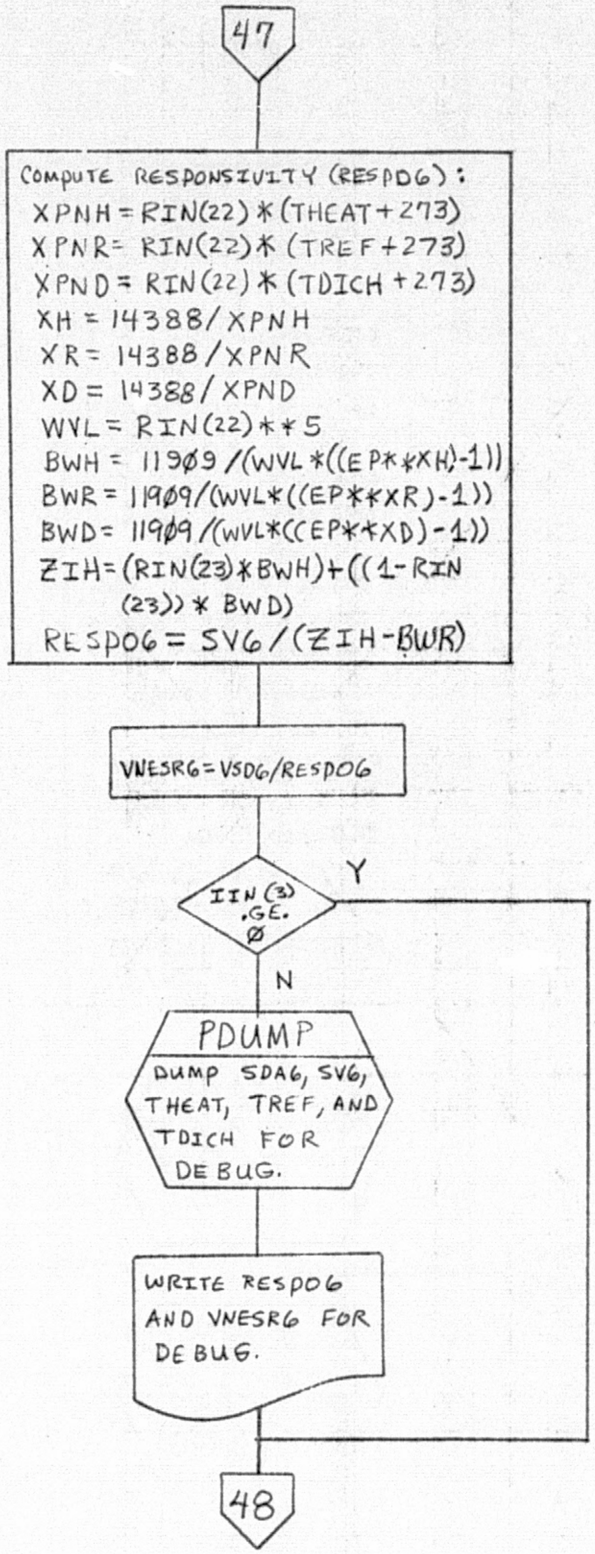
SUBTRACT CHANNEL BIAS VOLTAGE:
SV6 = VAVE6 - RIN(24)

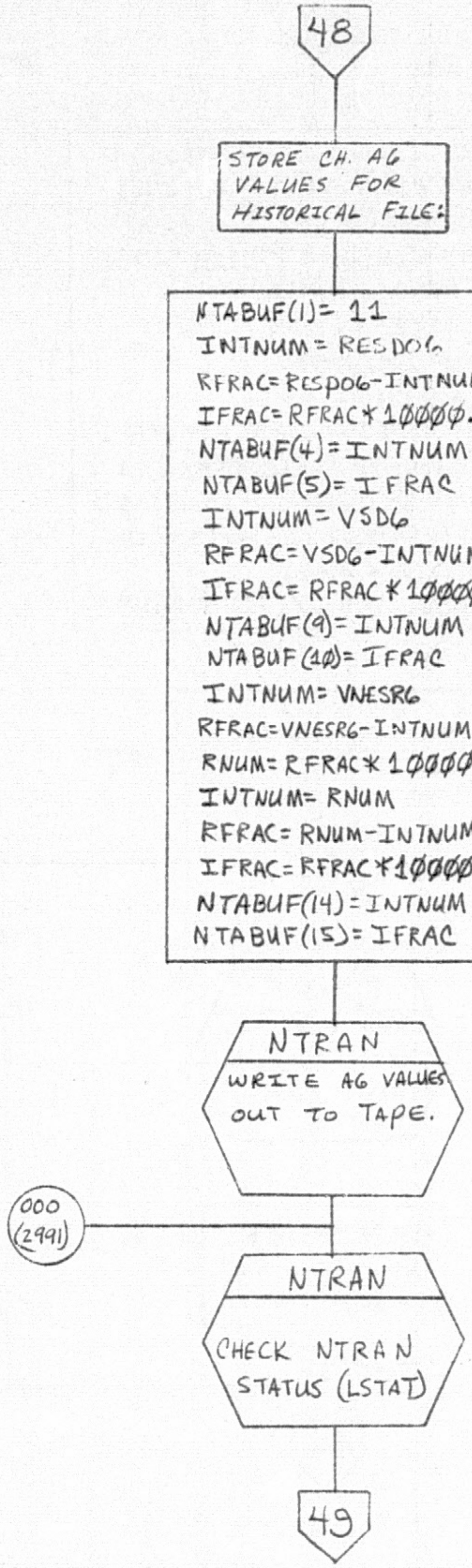
COMPUTE HEATED (THEAT), REFERENCE (TREF), AND DICHROIC (TDICH) TEMPS:
IHT = FLOAT(MEAS(376))
IRT = FLOAT(MEAS(484))
IDT = FLOAT(MEAS(322))
JJNT = 1
NDICAL = MEAS(322)
NTRCAL = MEAS(484)
THEAT = CNVTMP(1) + (CNVTMP(2) * IHT) + (CNVTMP(3) * (IHT ** 2)) + (CNVTMP(4) * (IHT ** 3)) + (CNVTMP(5) * (IHT ** 4)) + (CNVTMP(6) * (IHT ** 5))

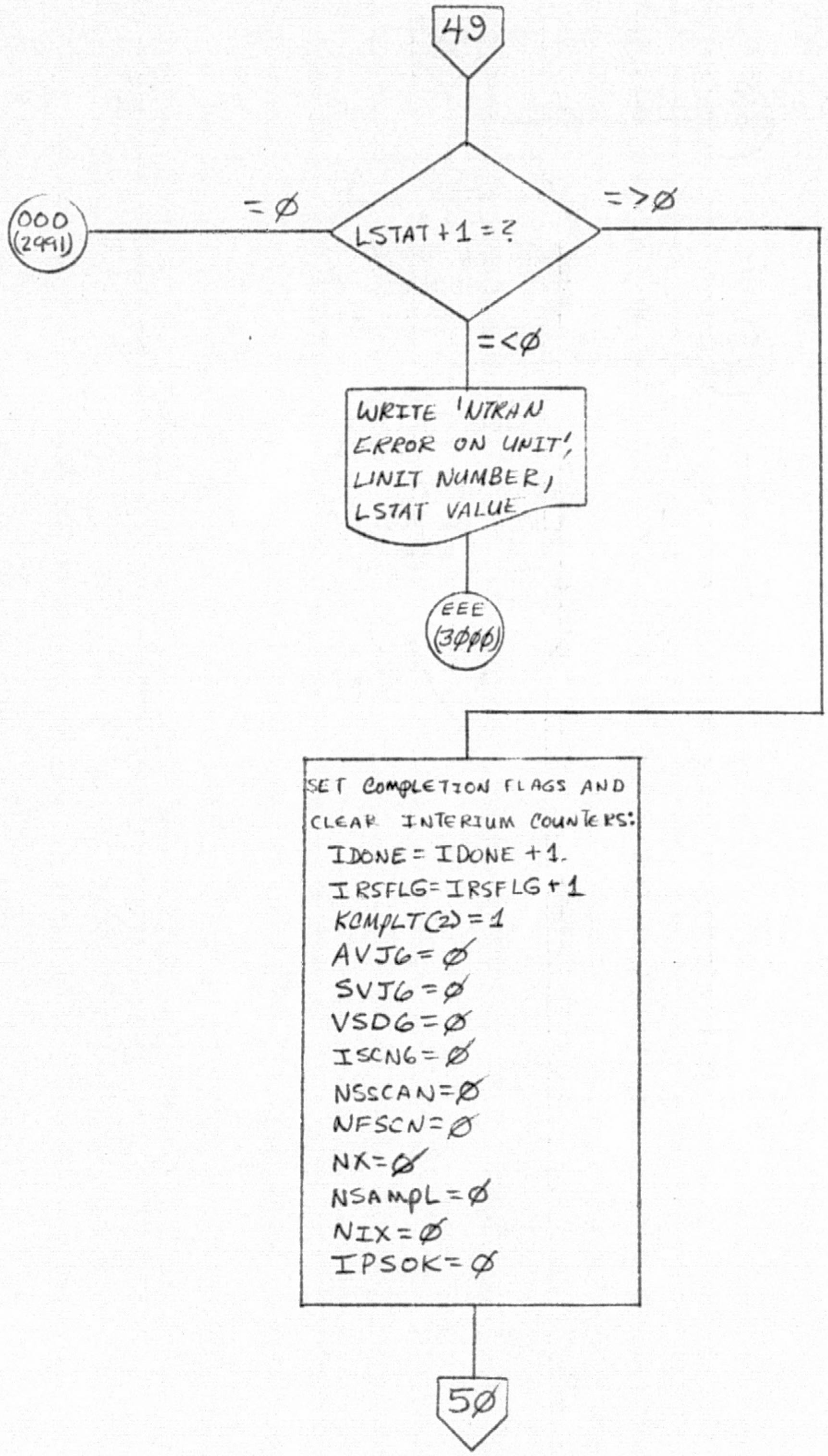
TREF = CNVTMP(37) + (CNVTMP(38) * IRT) + (CNVTMP(39) * (IRT ** 2)) + (CNVTMP(4Ø) * (IRT ** 3) + (CNVTMP(41) * (IRT ** 4)) + (CNVTMP(42) * (IRT ** 5))

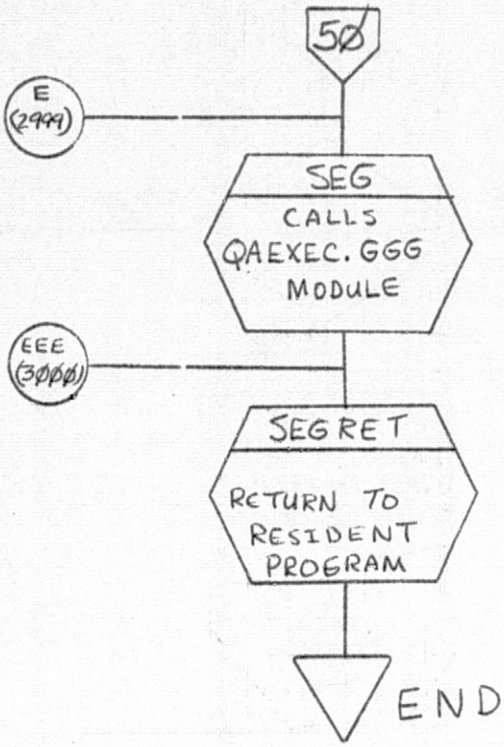
TDICH = CNVTMP(49) + (CNVTMP(5Ø) * IDT) + (CNVTMP(51) * (IDT ** 2)) + (CNVTMP(52) * (IDT ** 3)) + (CNVTMP(53) * (IDT ** 4)) + (CNVTMP(54) * (IDT ** 5))

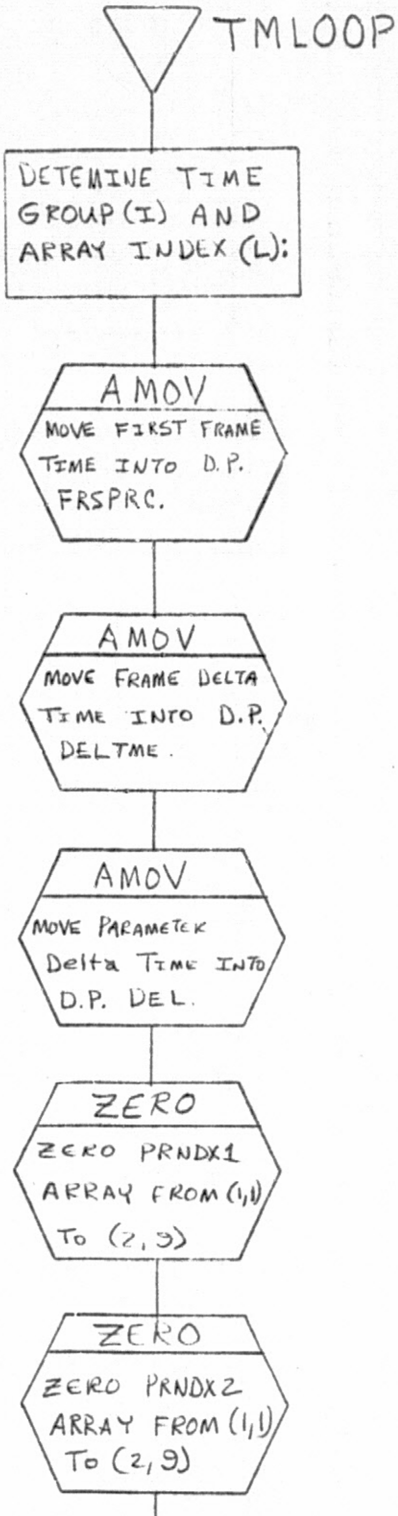
47

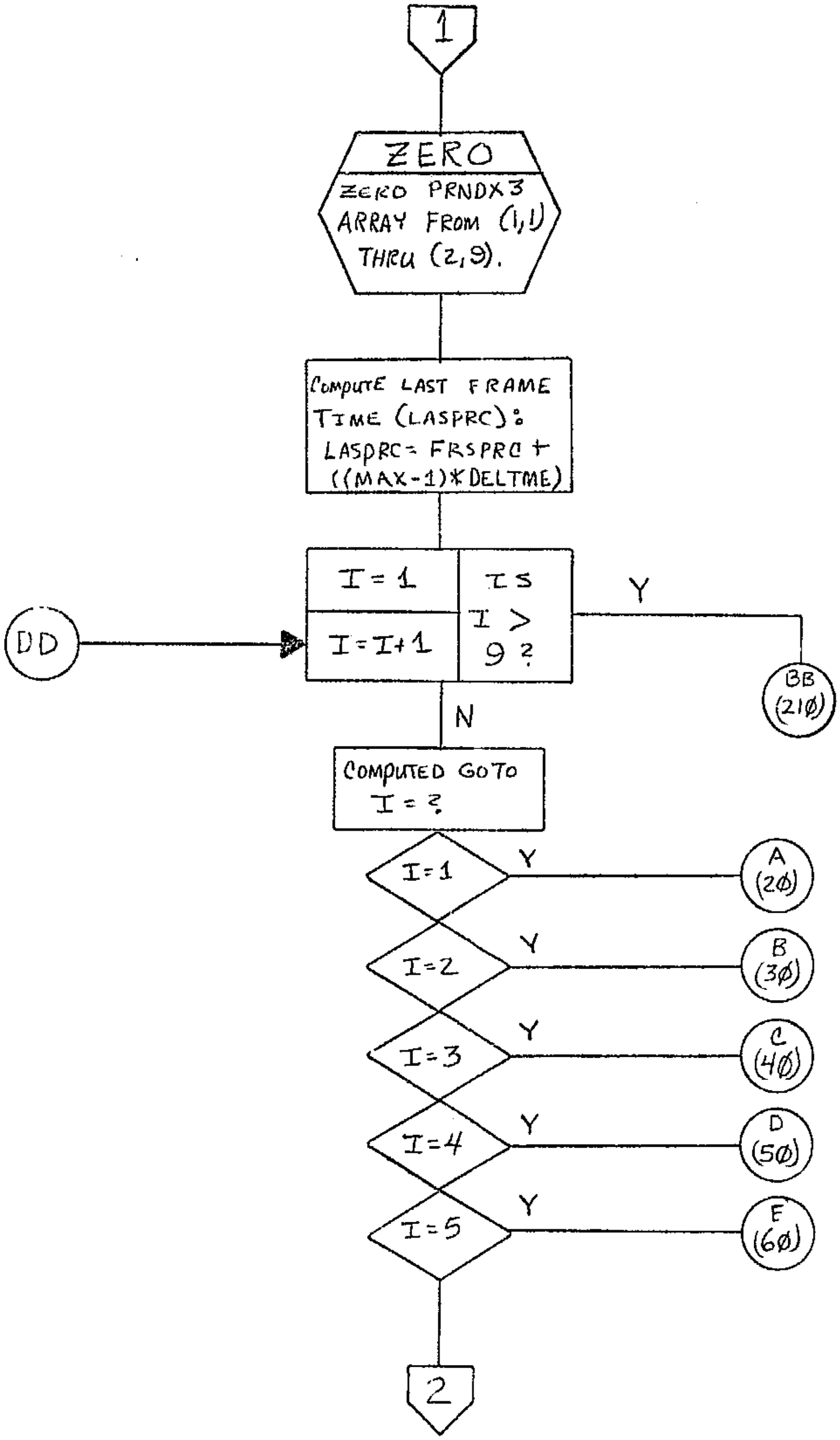

47
COMPUTE RESPONSIVITY (RESPD6):
XPNH = RIN(22) * (THEAT + 273)
XPNR = RIN(22) * (TREF + 273)
XPND = RIN(22) * (TDICH + 273)
XH = 14388/XPNH
XR = 14388/XPNR
XD = 14388/XPND
WVL = RIN(22) ** 5
BWH = 11909/(WVL*((EP**XH)-1))
BWR = 11909/(WVL*((EP**XR)-1))
BWD = 11909/(WVL*((EP**XD)-1))
ZIH = (RIN(23)*BWH)+((1-RIN(23))*BWD)
RESPO6 = SV6/(ZIH-BWR)
VNESR6 = VSD6/RESPO6
IIN(3) .GE. Ø
Y
N
PDUMP
DUMP SDA6, SV6, THEAT, TREF, AND TDICH FOR DEBUG.
WRITE RESPO6 AND VNESR6 FOR DEBUG.
48

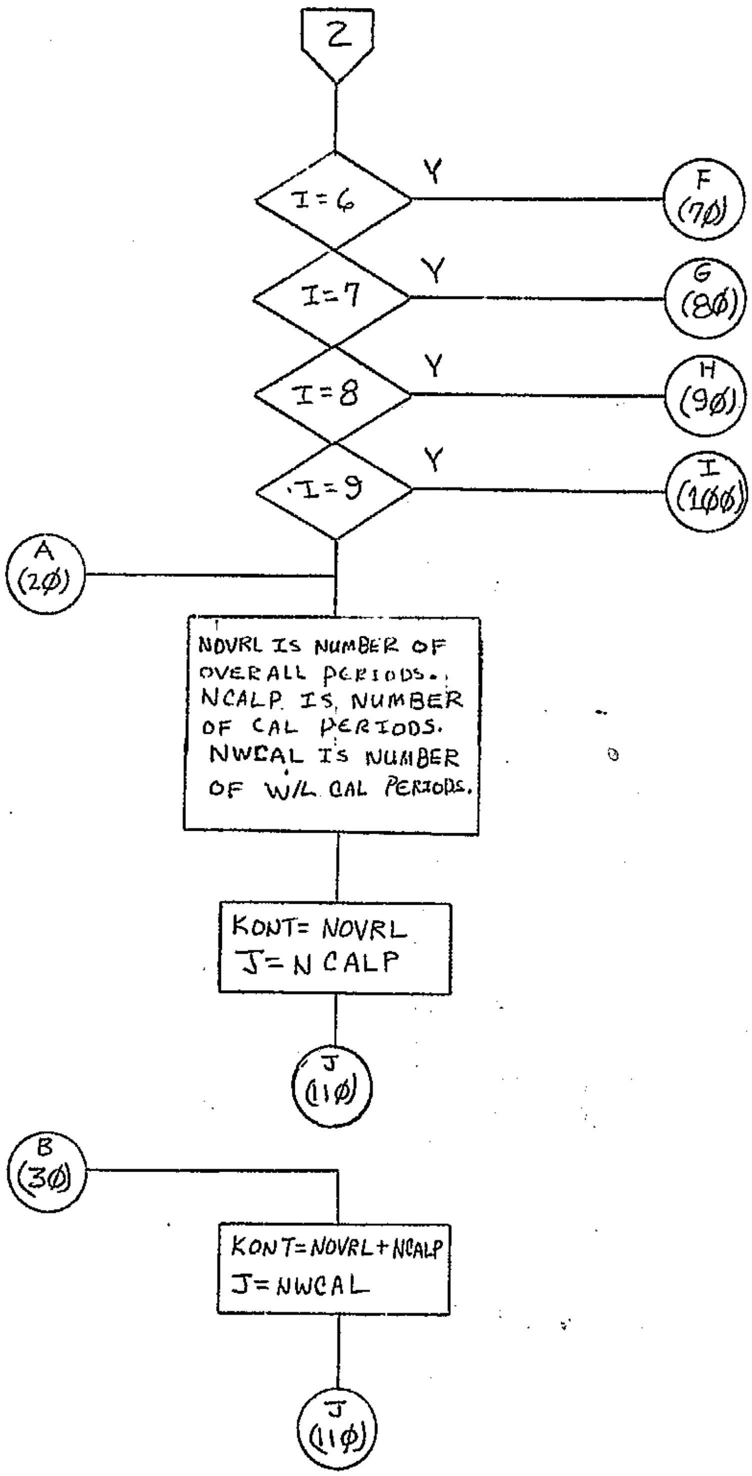

48
STORE CH. A6 VALUES FOR HISTORICAL FILE:
NTABUF(1)= 11
INTNUM = RESD06
RFRAC=RESD06-INTNUM
IFRAC=RFRAC*10000.
NTABUF(4)=INTNUM
NTABUF(5)= IFRAC
INTNUM= VSD6
RFRAC=VSD6-INTNUM
IFRAC= RFRAC*10000.
NTABUF(9)= INTNUM
NTABUF(10)= IFRAC
INTNUM= VNESR6
RFRAC=VNESR6-INTNUM
RNUM= RFRAC*10000.
INTNUM= RNUM
RFRAC= RNUM-INTNUM
IFRAC= RFRAC*10000.
NTABUF(14)=INTNUM
NTABUF(15)= IFRAC
NTRAN
WRITE A6 VALUES OUT TO TAPE.
000 (2991)
NTRAN
CHECK NTRAN STATUS (LSTAT)
49


49
LSTAT + 1 = ?
= Ø
000 (2991)
=>Ø
=<Ø
WRITE 'NTRAN ERROR ON UNIT', UNIT NUMBER, LSTAT VALUE
EEE (3ØØØ)
SET COMPLETION FLAGS AND CLEAR INTERIUM COUNTERS:
IDONE = IDONE + 1
IRSFLG = IRSFLG + 1
KOMPLT(2) = 1
AVJ6 = Ø
SVJ6 = Ø
VSD6 = Ø
ISCN6 = Ø
NSSCAN = Ø
NFSCN = Ø
NX = Ø
NSAMPL = Ø
NIX = Ø
IPSOK = Ø
5Ø

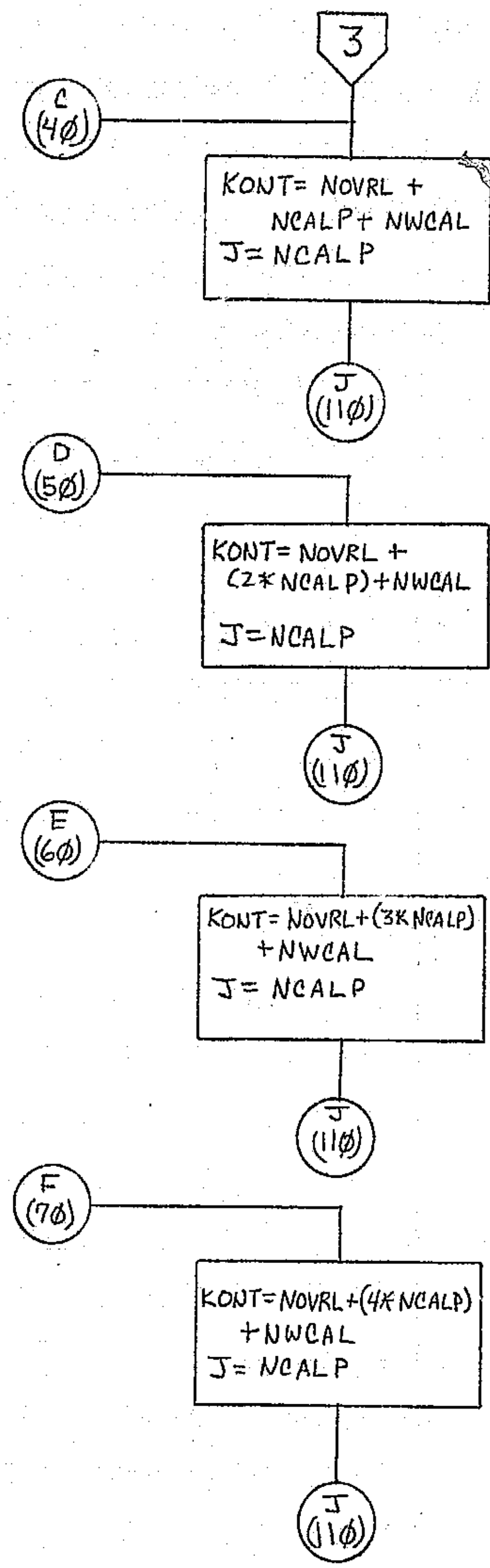

50
E
(2999)
SEG
CALLS
QAEXEC.GGG
MODULE
EEE
(3000)
SEGRET
RETURN TO
RESIDENT
PROGRAM
END

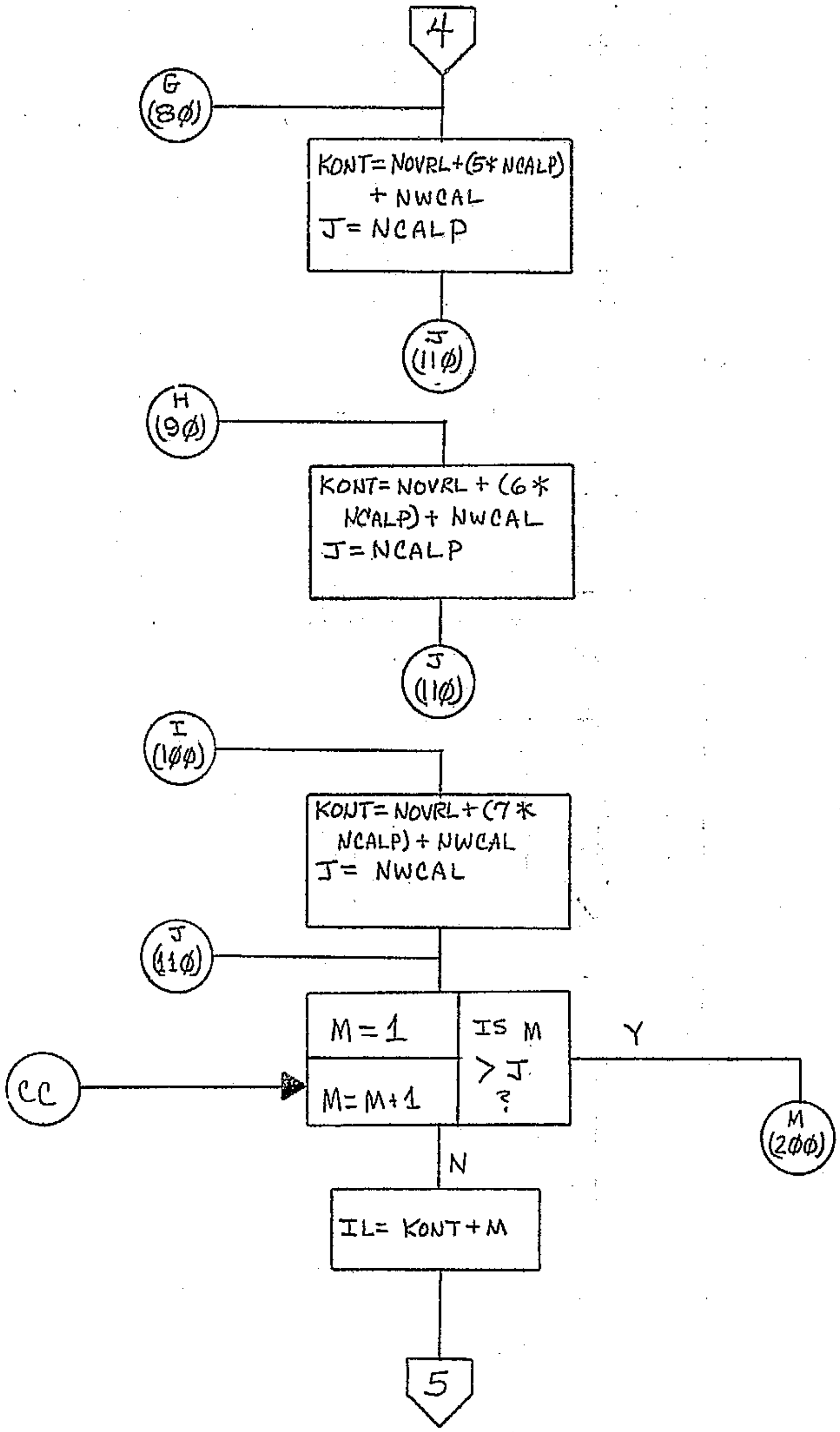

TMLOOP
DETEMINE TIME GROUP (I) AND ARRAY INDEX (L):
AMOV
MOVE FIRST FRAME TIME INTO D.P. FRSPRC.
AMOV
MOVE FRAME DELTA TIME INTO D.P. DELTME.
AMOV
MOVE PARAMETER Delta TIME INTO D.P. DEL.
ZERO
ZERO PRNDX1 ARRAY FROM (1,1) TO (2,9)
ZERO
ZERO PRNDX2 ARRAY FROM (1,1) TO (2,9)
1

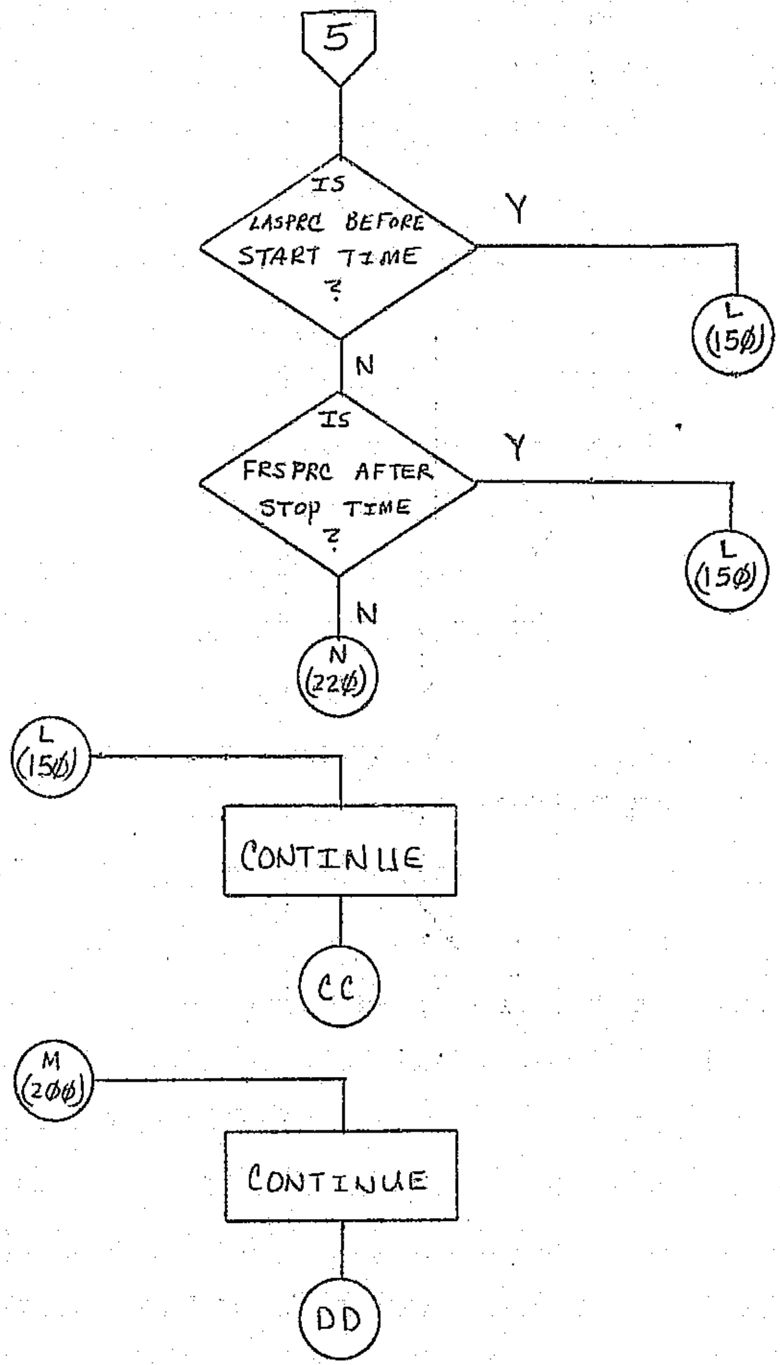

1
ZERO
ZERO PRNDX3 ARRAY FROM (1,1) THRU (2,9).
COMPUTE LAST FRAME TIME (LASPRC):
LASPRC = FRSPRC + ((MAX-1)*DELTME)
I = 1
I = I+1
IS I > 9 ?
Y
BB (21Ø)
DD
N
COMPUTED GOTO I = ?
I=1
Y
A (2Ø)
I=2
Y
B (3Ø)
I=3
Y
C (4Ø)
I=4
Y
D (5Ø)
I=5
Y
E (6Ø)
2

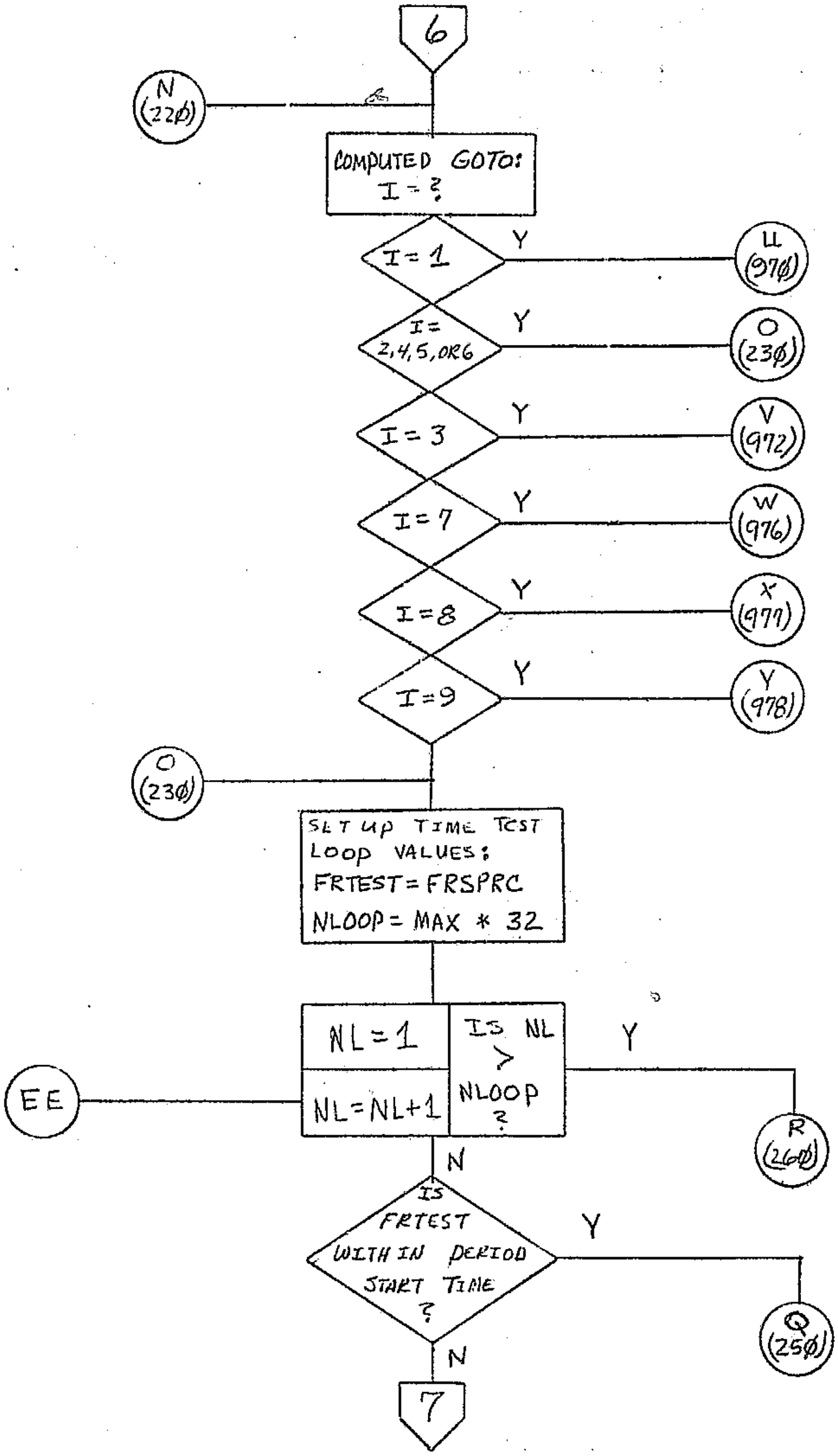


2
I=6
Y
F
(70)
I=7
Y
G
(80)
I=8
Y
H
(90)
I=9
Y
I
(100)
A
(20)
NOVRL IS NUMBER OF OVERALL PERIODS.
NCALP IS NUMBER OF CAL PERIODS.
NWCAL IS NUMBER OF W/L CAL PERIODS.
KONT= NOVRL
J= N CALP
J
(110)
B
(30)
KONT=NOVRL+NCALP
J=NWCAL
J
(110)

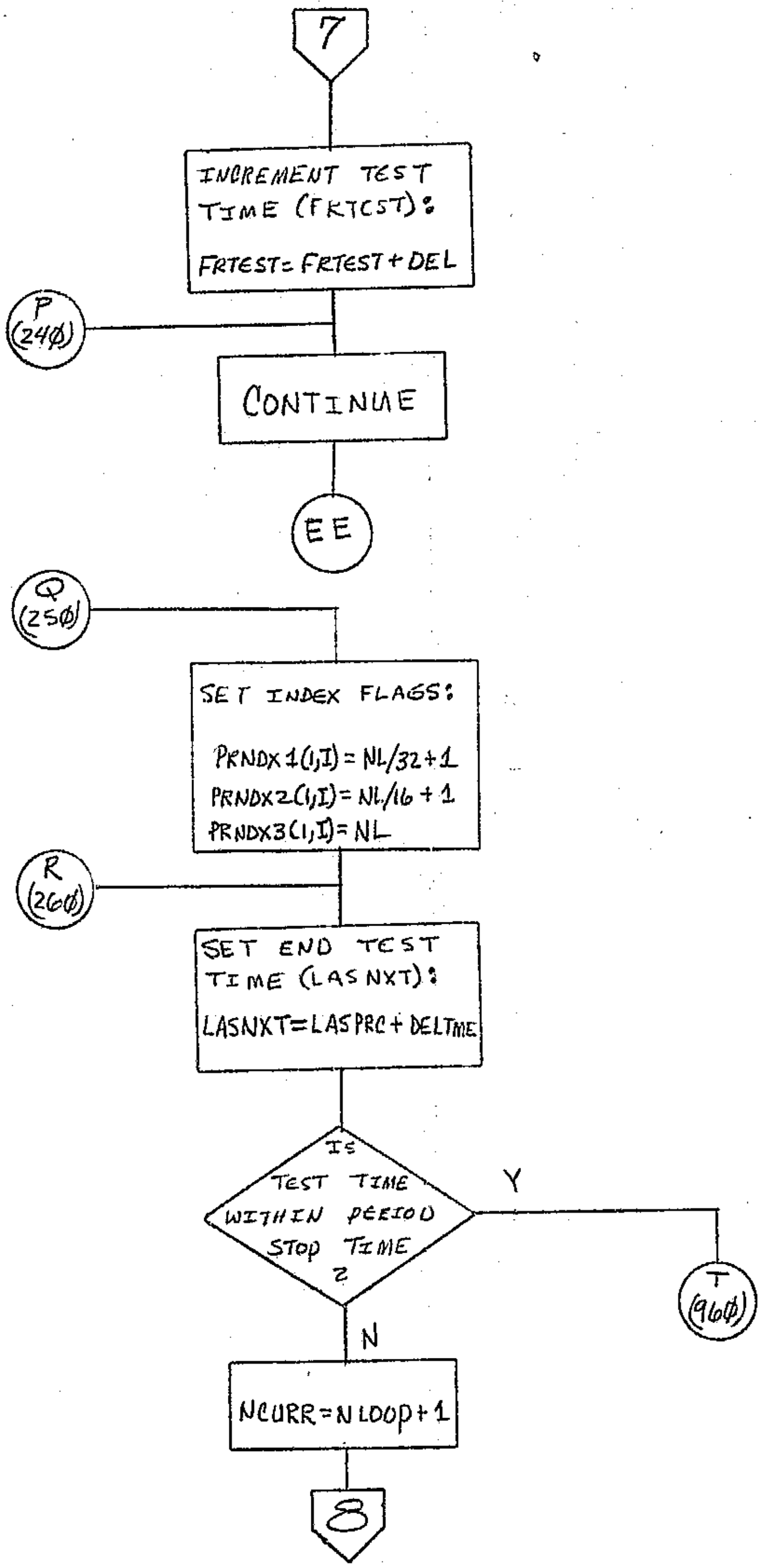

3
C (4Ø)
KONT= NOVRL + NCALP+ NWCAL
J= NCALP
J (11Ø)
D (5Ø)
KONT= NOVRL + (2* NCALP)+NWCAL
J=NCALP
J (11Ø)
E (6Ø)
KONT= NOVRL+(3*NCALP) +NWCAL
J= NCALP
J (11Ø)
F (7Ø)
KONT=NOVRL+(4*NCALP) +NWCAL
J= NCALP
J (11Ø)

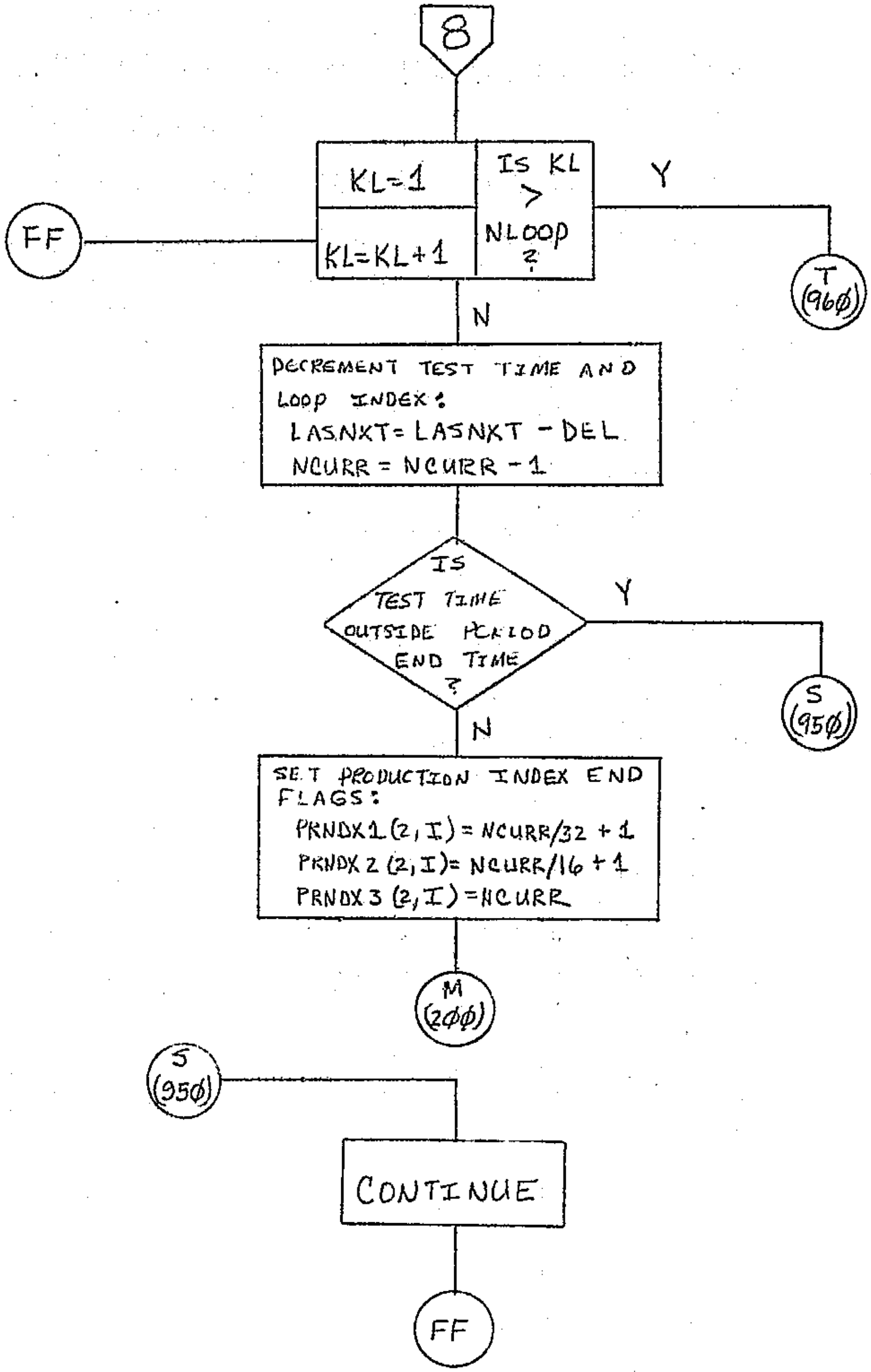

4
G (8Ø)
KONT = NOVRL + (5 * NCALP) + NWCAL
J = NCALP
J (11Ø)
H (9Ø)
KONT = NOVRL + (6 * NCALP) + NWCAL
J = NCALP
J (11Ø)
I (1ØØ)
KONT = NOVRL + (7 * NCALP) + NWCAL
J = NWCAL
J (11Ø)
M = 1
M = M+1
IS M > J ?
Y
M (2ØØ)
CC
N
IL = KONT + M
5

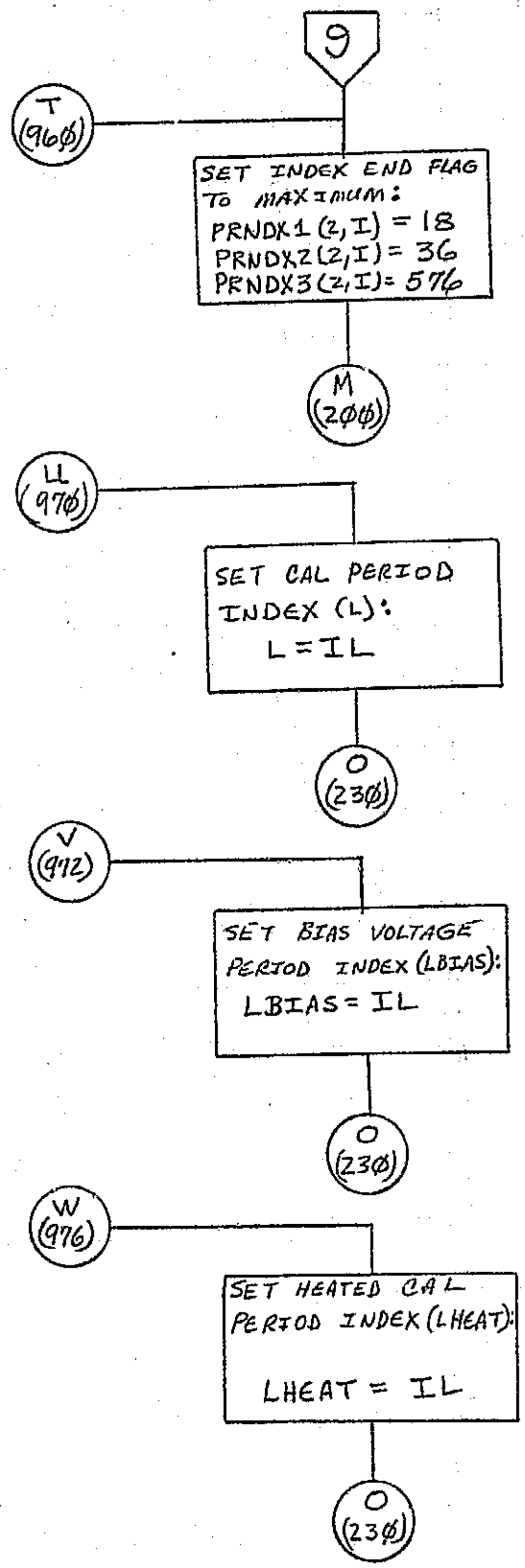

5
IS LASPRC BEFORE START TIME ?
Y
L (15Ø)
N
IS FRSPRC AFTER STOP TIME ?
Y
L (15Ø)
N
N (22Ø)
L (15Ø)
CONTINUE
CC
M (2ØØ)
CONTINUE
DD

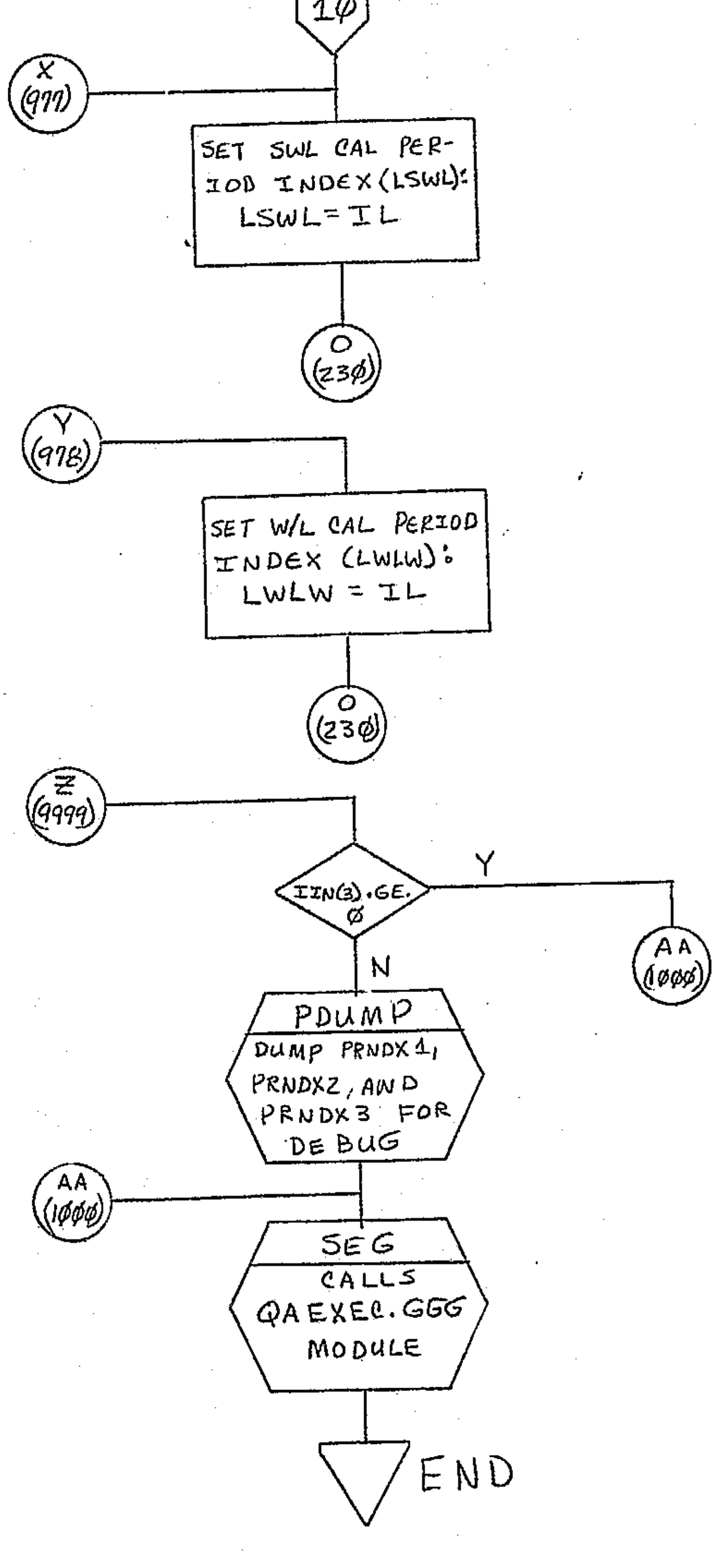


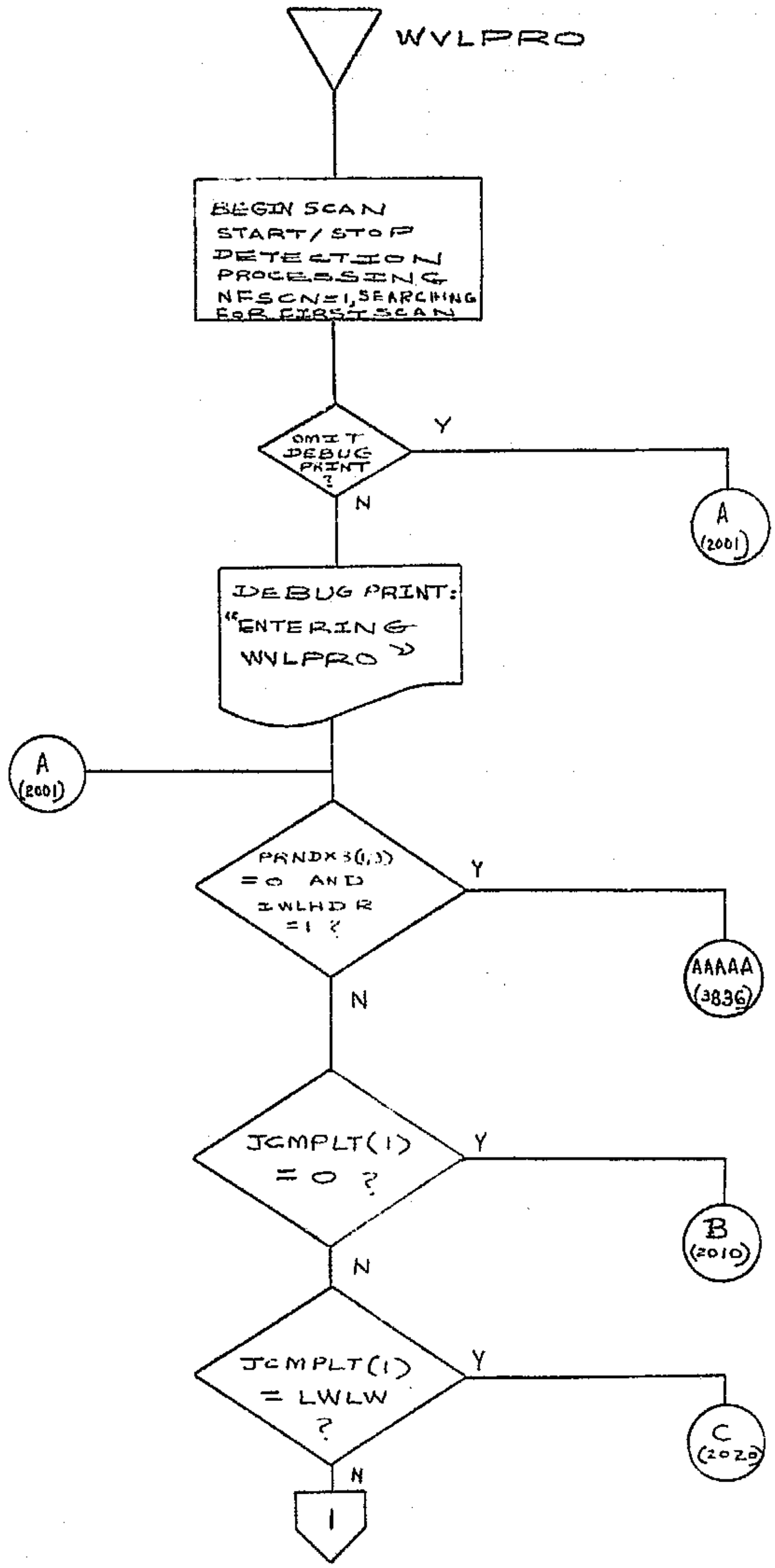
6
N (22Ø)
COMPUTED GOTO: I = ?
I = 1
Y
U (97Ø)
I = 2,4,5,OR6
Y
O (23Ø)
I = 3
Y
V (972)
I = 7
Y
W (976)
I = 8
Y
X (977)
I = 9
Y
Y (978)
O (23Ø)
SET UP TIME TEST LOOP VALUES:
FRTEST = FRSPRC
NLOOP = MAX * 32
NL = 1
NL = NL+1
IS NL > NLOOP ?
Y
R (26Ø)
EE
N
IS FRTEST WITH IN PERIOD START TIME ?
Y
Q (25Ø)
N
7

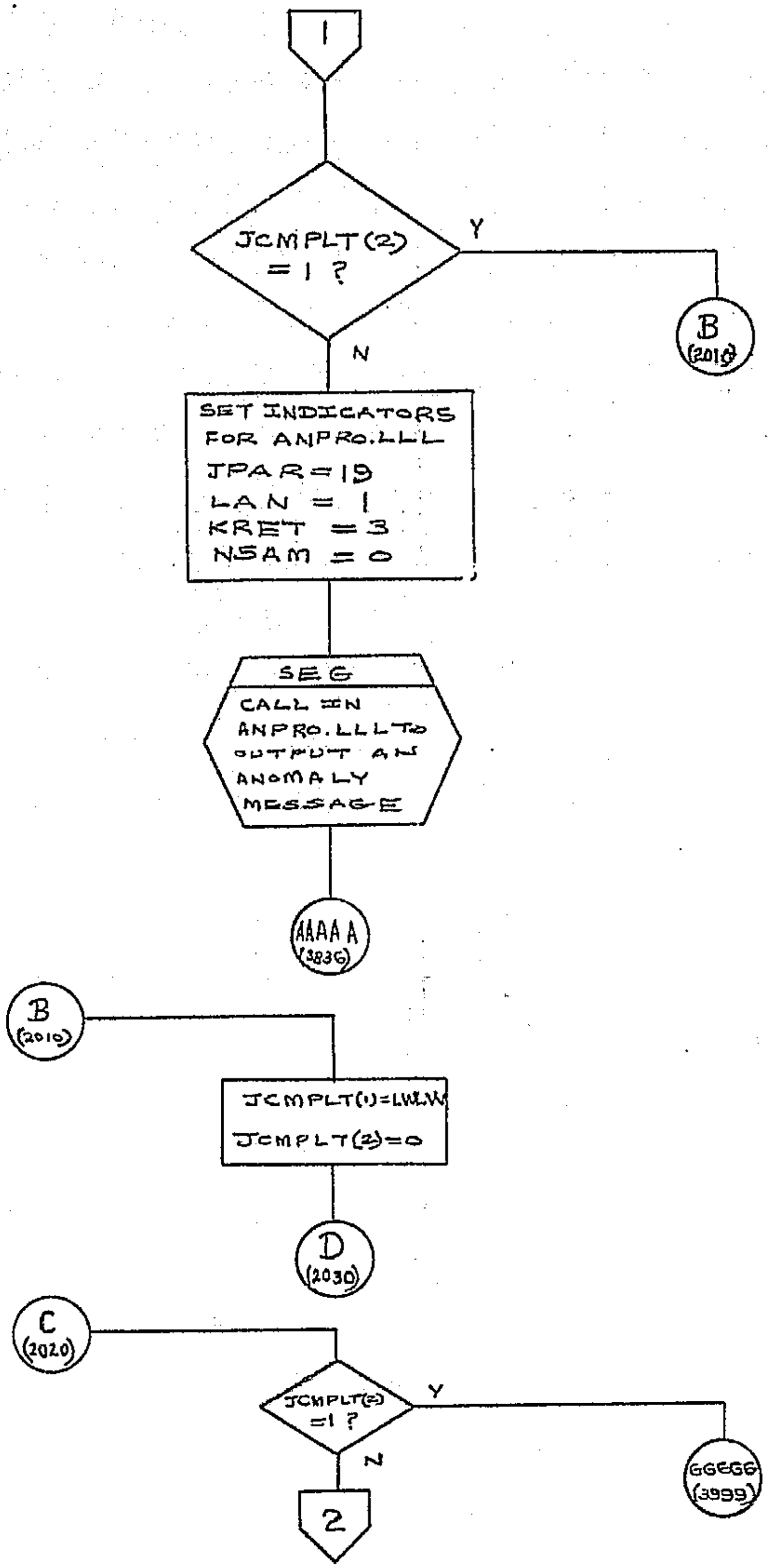

7
INCREMENT TEST TIME (FRTEST):
FRTEST= FRTEST + DEL
P (240)
CONTINUE
EE
Q (250)
SET INDEX FLAGS:
PRNDX1(1,I) = NL/32+1
PRNDX2(1,I) = NL/16 + 1
PRNDX3(1,I) = NL
R (260)
SET END TEST TIME (LASNXT):
LASNXT=LASPRC+DELTME
IS TEST TIME WITHIN PERIOD STOP TIME ?
Y
T (960)
N
NCURR=NLOOP+1
8

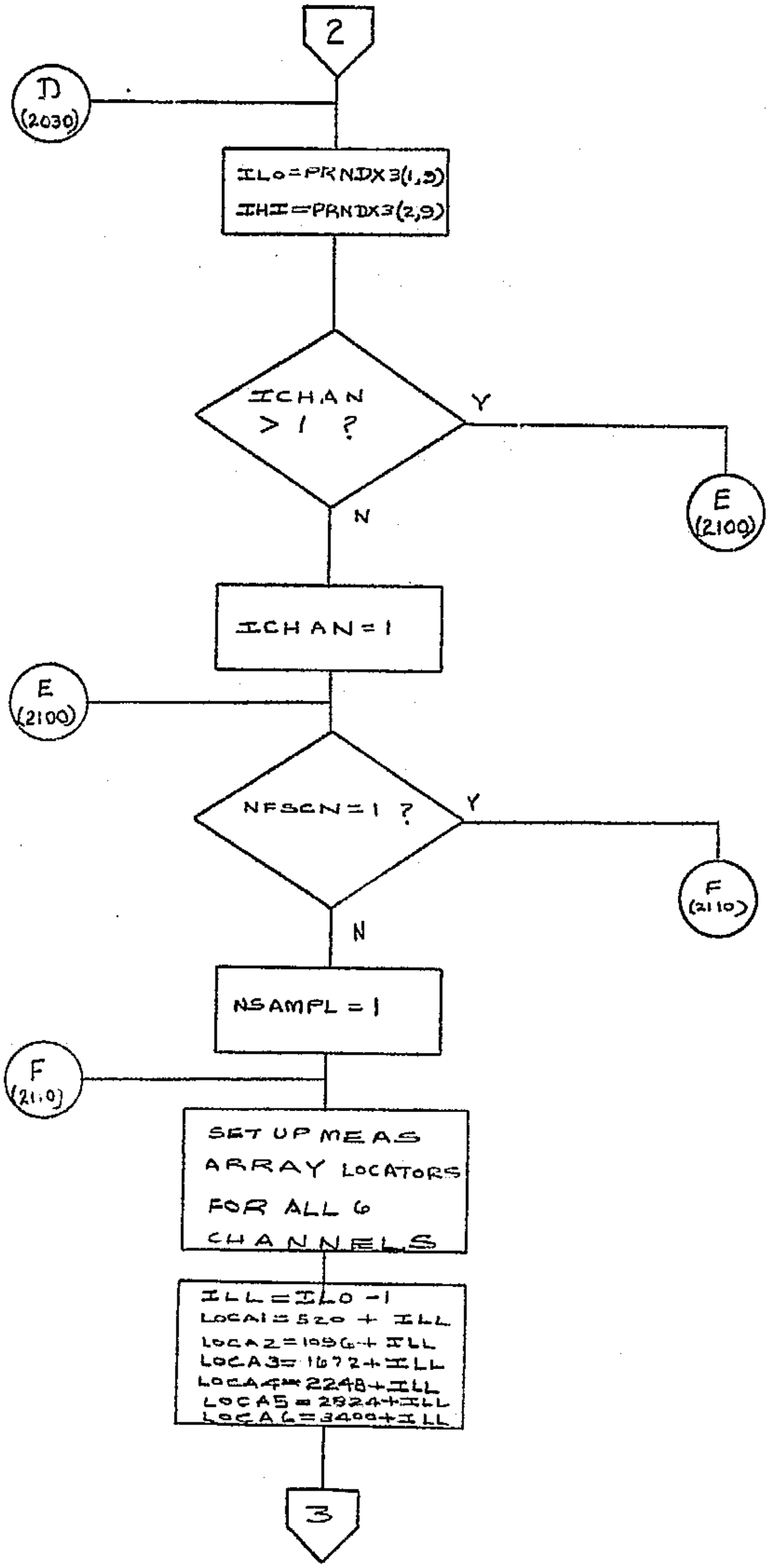

8
KL=1
KL=KL+1
IS KL > NLOOP ?
Y
FF
T (96Ø)
N
DECREMENT TEST TIME AND LOOP INDEX:
LASNKT= LASNKT - DEL
NCURR = NCURR - 1
IS TEST TIME OUTSIDE PERIOD END TIME ?
Y
S (95Ø)
N
SET PRODUCTION INDEX END FLAGS:
PRNDX1 (2, I) = NCURR/32 + 1
PRNDX2 (2, I) = NCURR/16 + 1
PRNDX3 (2, I) = NCURR
M (2ØØ)
S (95Ø)
CONTINUE
FF

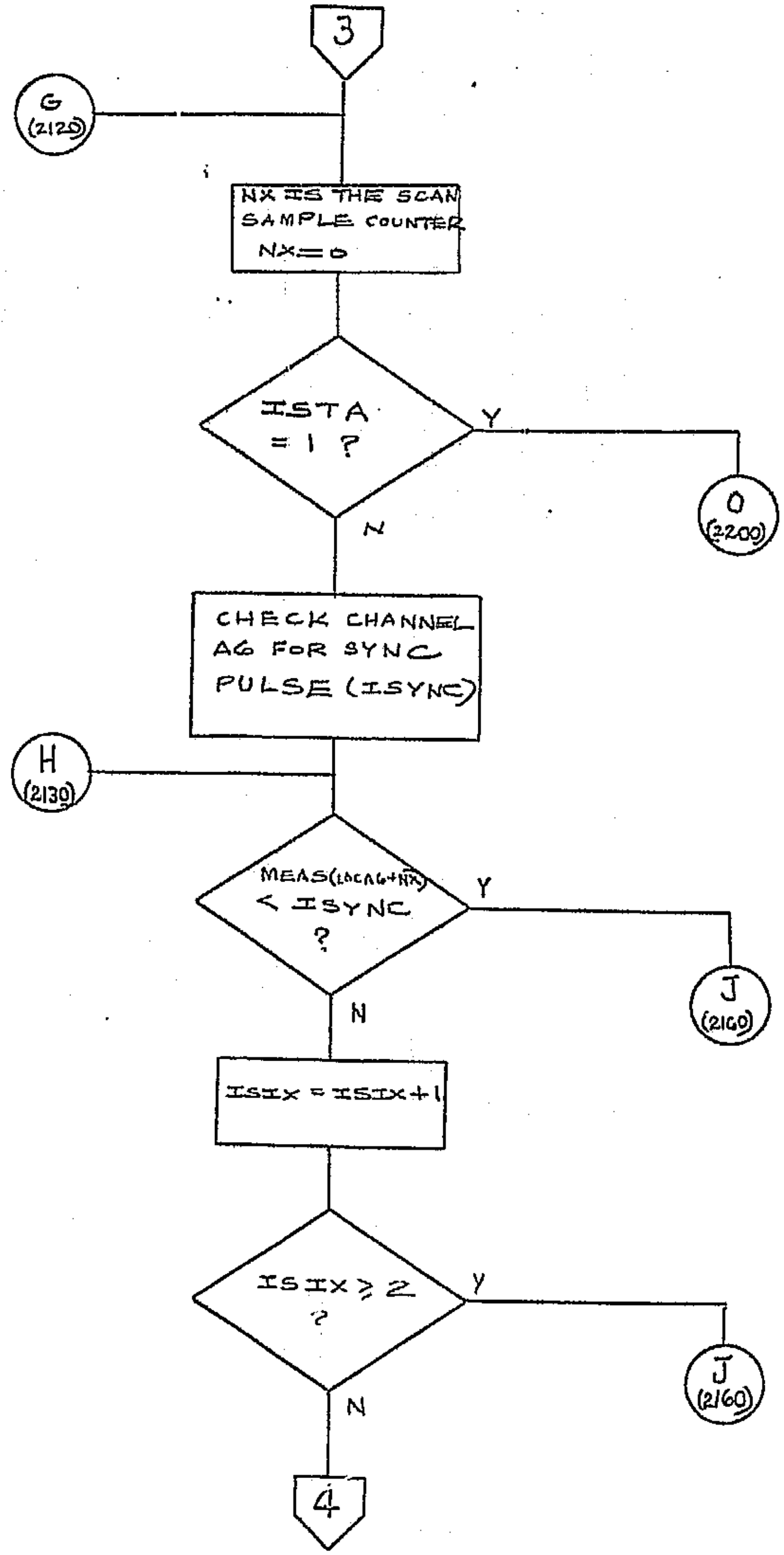

9
T (96Ø)
SET INDEX END FLAG TO MAXIMUM:
PRNDX1 (2,I) = 18
PRNDX2 (2,I) = 36
PRNDX3 (2,I) = 576
M (2ØØ)
U (97Ø)
SET CAL PERIOD INDEX (L):
L = IL
O (23Ø)
V (972)
SET BIAS VOLTAGE PERIOD INDEX (LBIAS):
LBIAS = IL
O (23Ø)
W (976)
SET HEATED CAL PERIOD INDEX (LHEAT):
LHEAT = IL
O (23Ø)

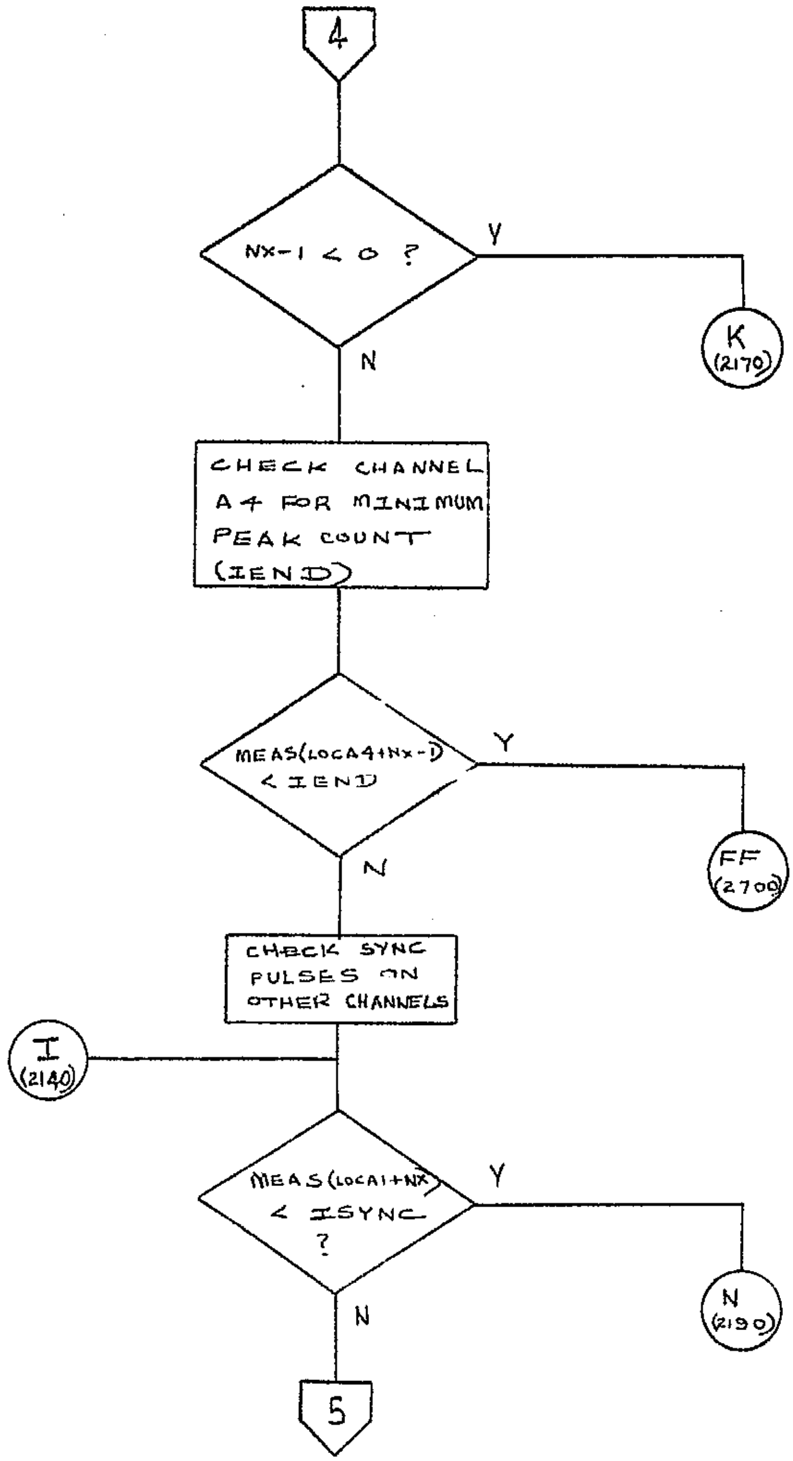

10
X (977)
SET SWL CAL PERIOD INDEX (LSWL): LSWL = IL
O (230)
Y (978)
SET W/L CAL PERIOD INDEX (LWLW): LWLW = IL
O (230)
Z (9999)
IIN(3).GE.0
Y
AA (1000)
N
PDUMP
DUMP PRNDX1, PRNDX2, AND PRNDX3 FOR DEBUG
AA (1000)
SEG
CALLS QAEXEC.GGG MODULE
END

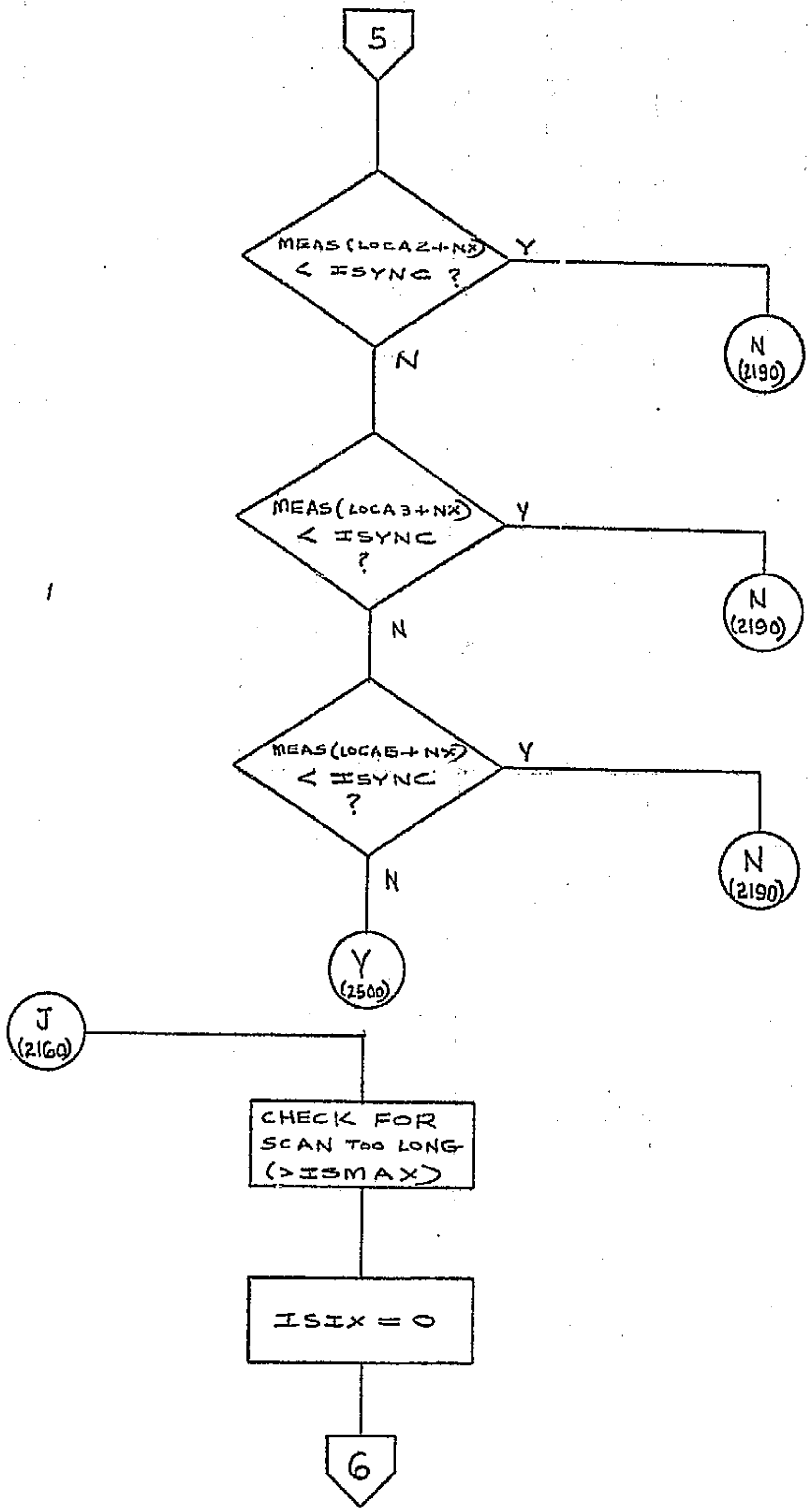

WVLPRO
BEGIN SCAN START/STOP DETECTION PROCESSING NFSCN=1, SEARCHING FOR FIRST SCAN
OMIT DEBUG PRINT ?
Y
N
A (2001)
DEBUG PRINT: "ENTERING WVLPRO"
A (2001)
PRNDX3(1,3) =0 AND IWLHDR =1 ?
Y
AAAAA (3836)
N
JCMPLT(1) = 0 ?
Y
B (2010)
N
JCMPLT(1) = LWLW ?
Y
C (2020)
N
1

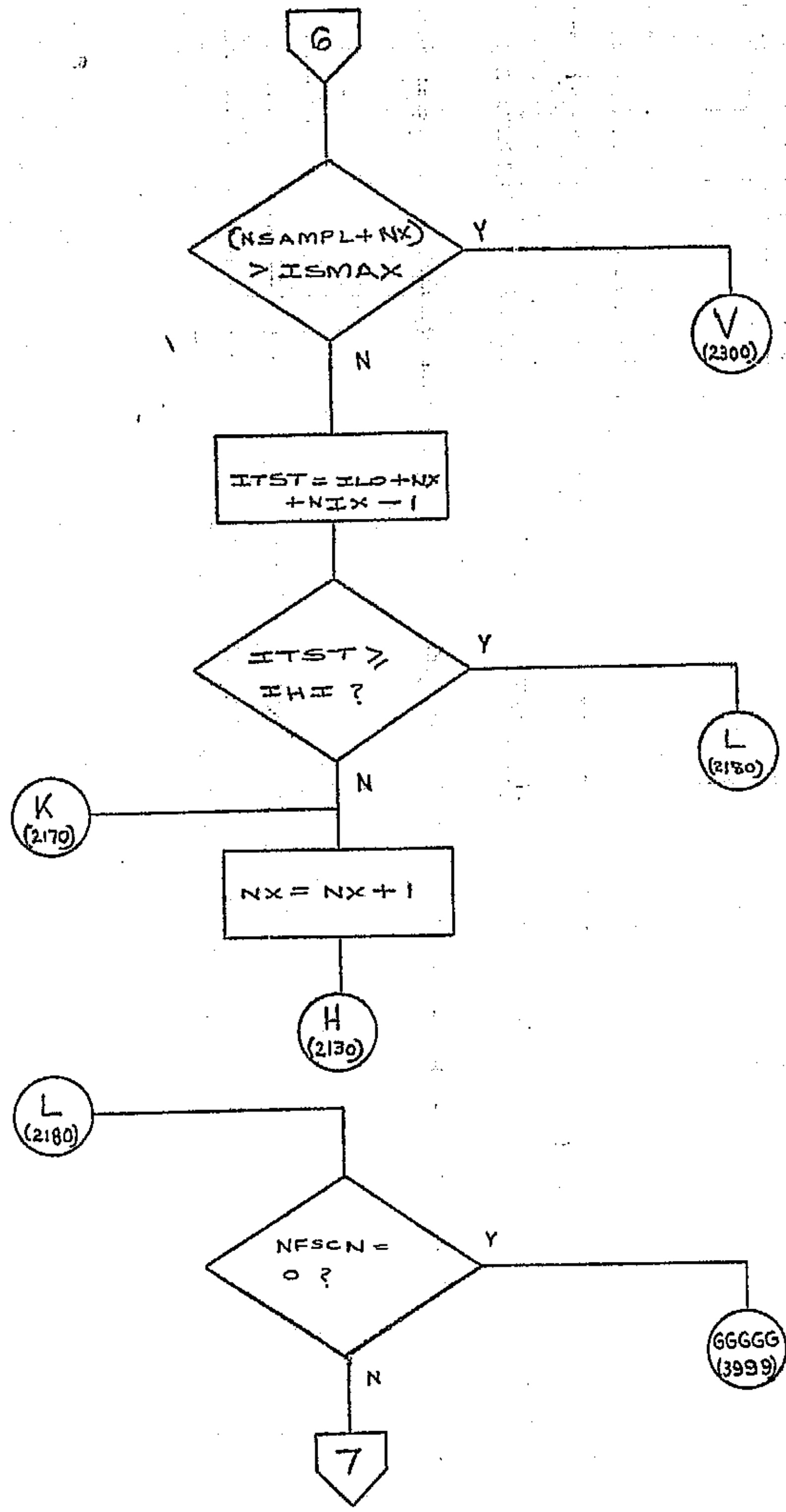

1
JCMPLT(2) =1 ?
Y
N
B (2010)
SET INDICATORS FOR ANPRO.LLL
JPAR=19
LAN = 1
KRET =3
NSAM =0
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
AAAAA (3836)
B (2010)
JCMPLT(1)=LWLW
JCMPLT(2)=0
D (2030)
C (2020)
JCMPLT(2) =1 ?
Y
N
GGEGG (3999)
2

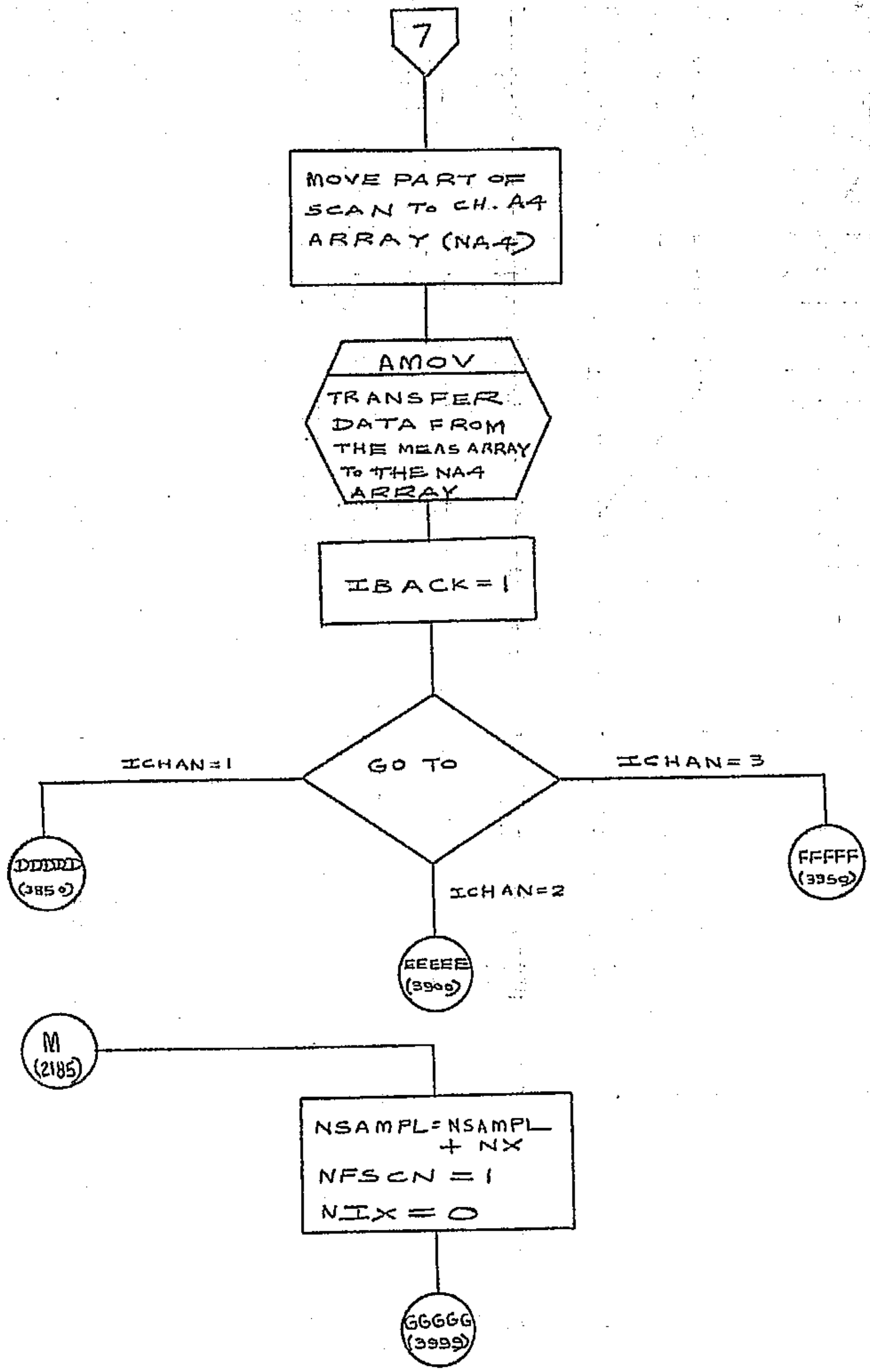

2
D
(2030)
ILO=PRNDX3(1,9)
IHI=PRNDX3(2,9)
ICHAN
> 1 ?
Y
E
(2100)
N
ICHAN=1
E
(2100)
NFSCN=1 ?
Y
F
(2110)
N
NSAMPL = 1
F
(2110)
SET UP MEAS
ARRAY LOCATORS
FOR ALL 6
CHANNELS
ILL=ILO -1
LOCA1=520 + ILL
LOCA2=1096+ ILL
LOCA3=1672+ILL
LOCA4=2248+ILL
LOCA5=2824+ILL
LOCA6=3400+ILL
3

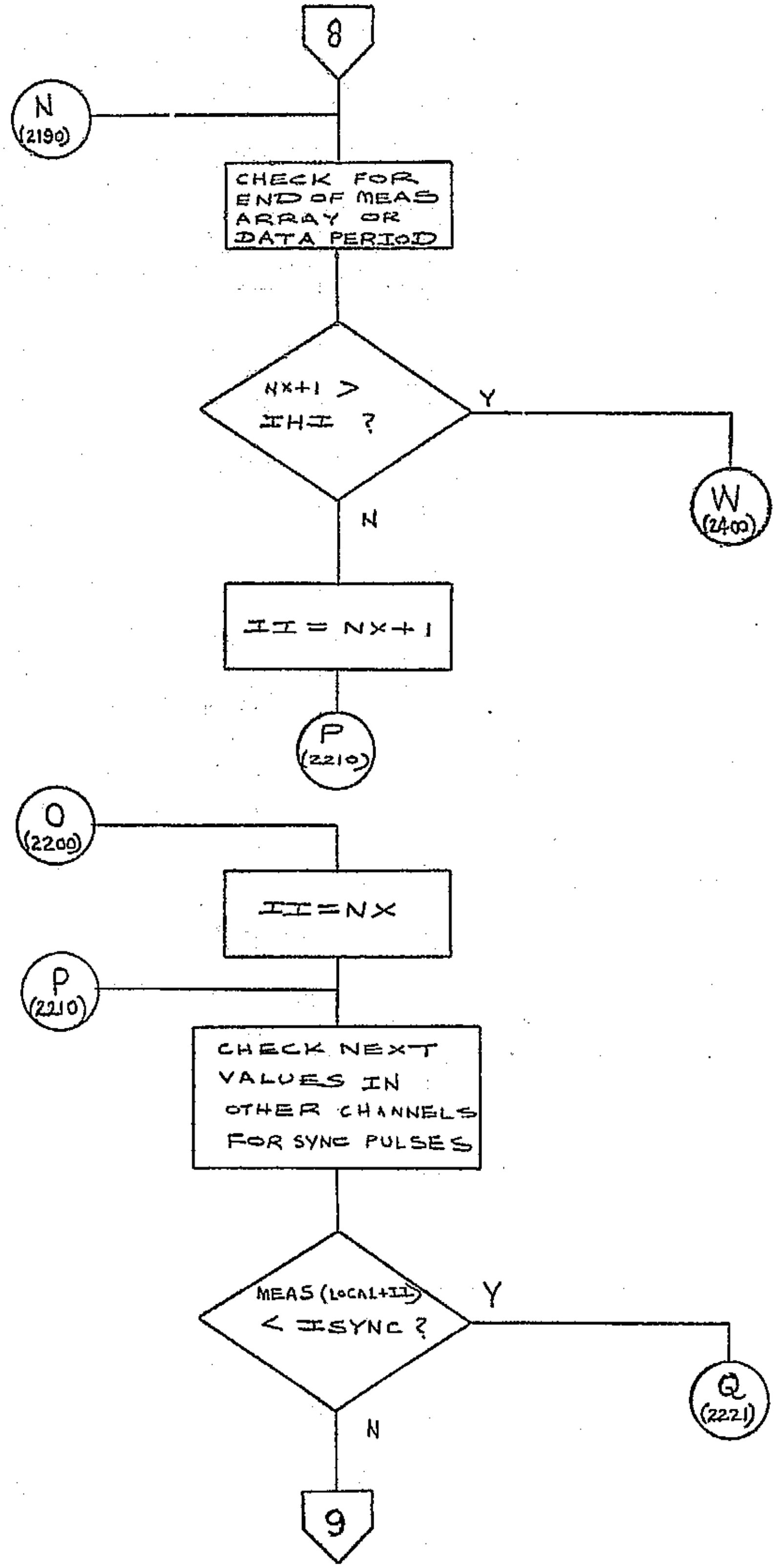

3
G
(2120)
NX IS THE SCAN SAMPLE COUNTER
NX = 0
ISTA = 1 ?
Y
N
O
(2200)
CHECK CHANNEL A6 FOR SYNC PULSE (ISYNC)
H
(2130)
MEAS(LOCA6+NX) < ISYNC ?
Y
N
J
(2160)
ISIX = ISIX + 1
ISIX ≥ 2 ?
Y
N
J
(2160)
4

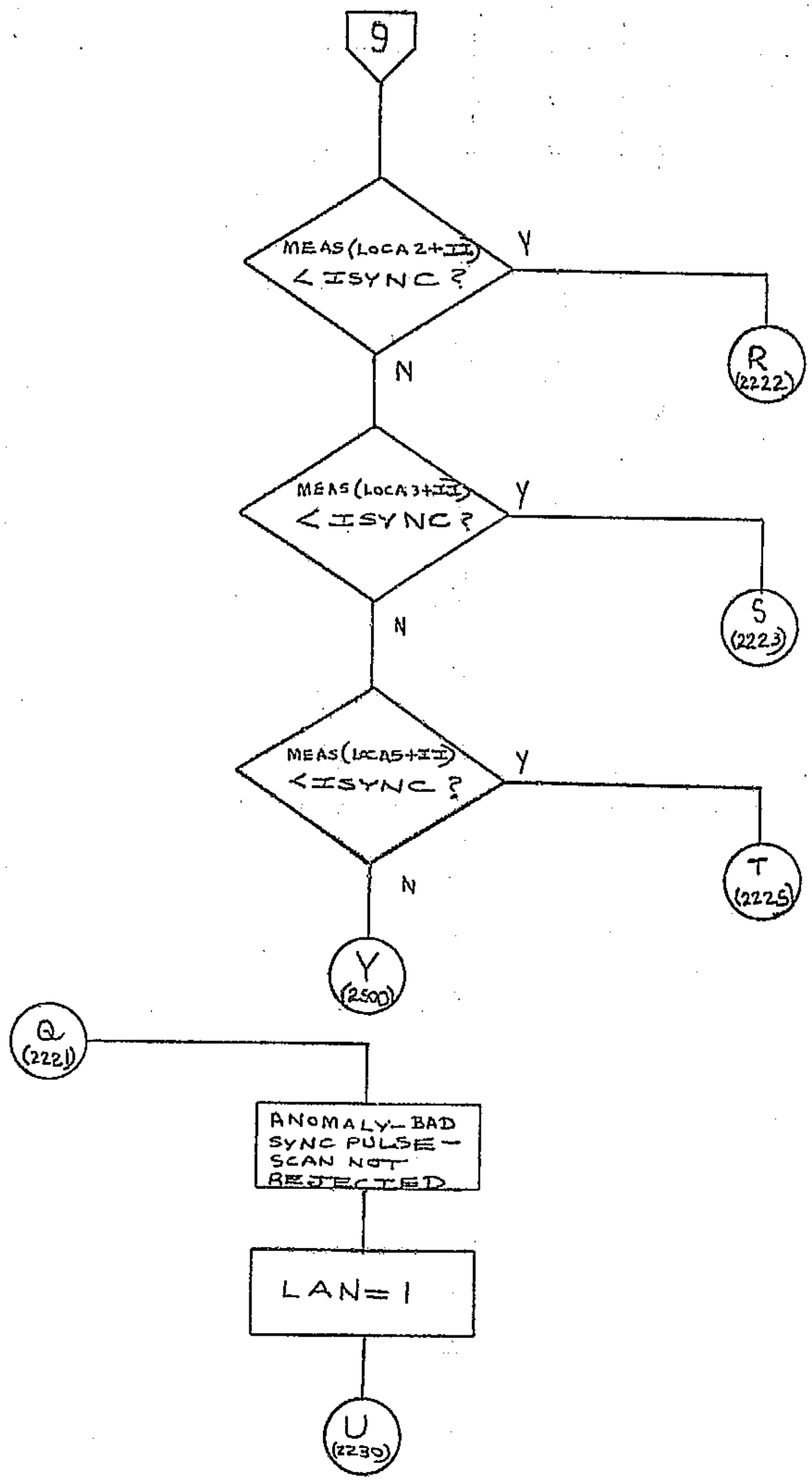

4
NX-1 < 0 ?
Y
K
(2170)
N
CHECK CHANNEL
A4 FOR MINIMUM
PEAK COUNT
(IEND)
MEAS(LOCA4+NX-1)
< IEND
Y
FF
(2700)
N
CHECK SYNC
PULSES ON
OTHER CHANNELS
I
(2140)
MEAS(LOCA1+NX)
< ISYNC
?
Y
N
(2190)
N
5

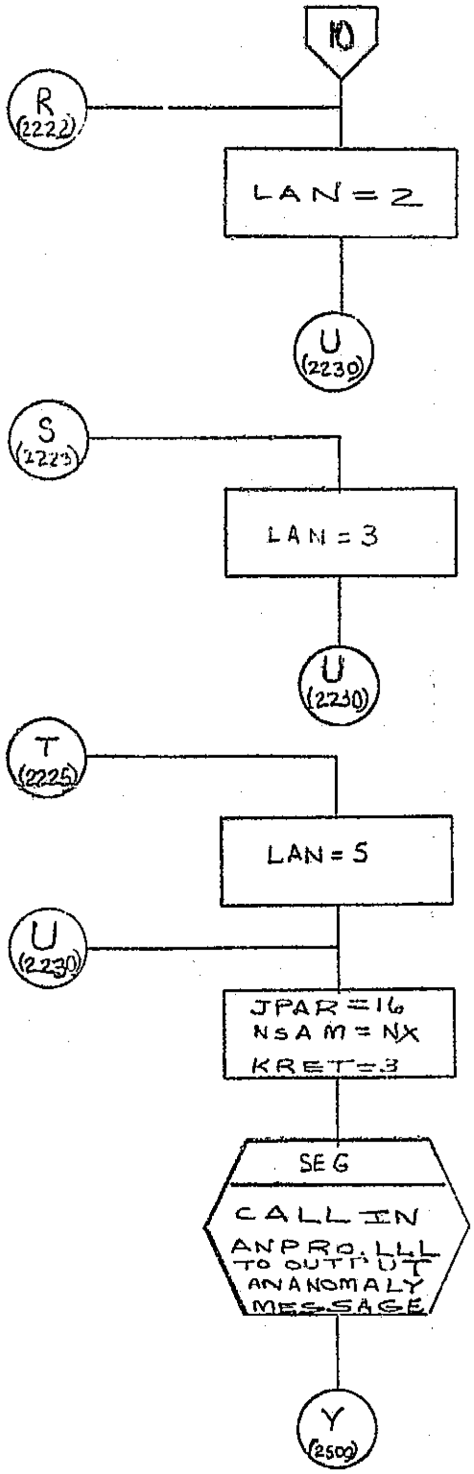

5
MEAS(LOCA2+NX) < ISYNC ?
Y
N
N (2190)
MEAS(LOCA3+NX) < ISYNC ?
Y
N
N (2190)
MEAS(LOCA5+NX) < ISYNC ?
Y
N
N (2190)
Y (2500)
J (2160)
CHECK FOR SCAN TOO LONG (>ISMAX)
ISIX = 0
6


6
(NSAMPL+NX) > ISMAX
Y
V (2300)
N
ITST = ILO+NX +NIX − 1
ITST ≥ IHI ?
Y
L (2180)
N
K (2170)
NX = NX + 1
H (2130)
L (2180)
NFSCN = 0 ?
Y
GGGGG (3999)
N
7

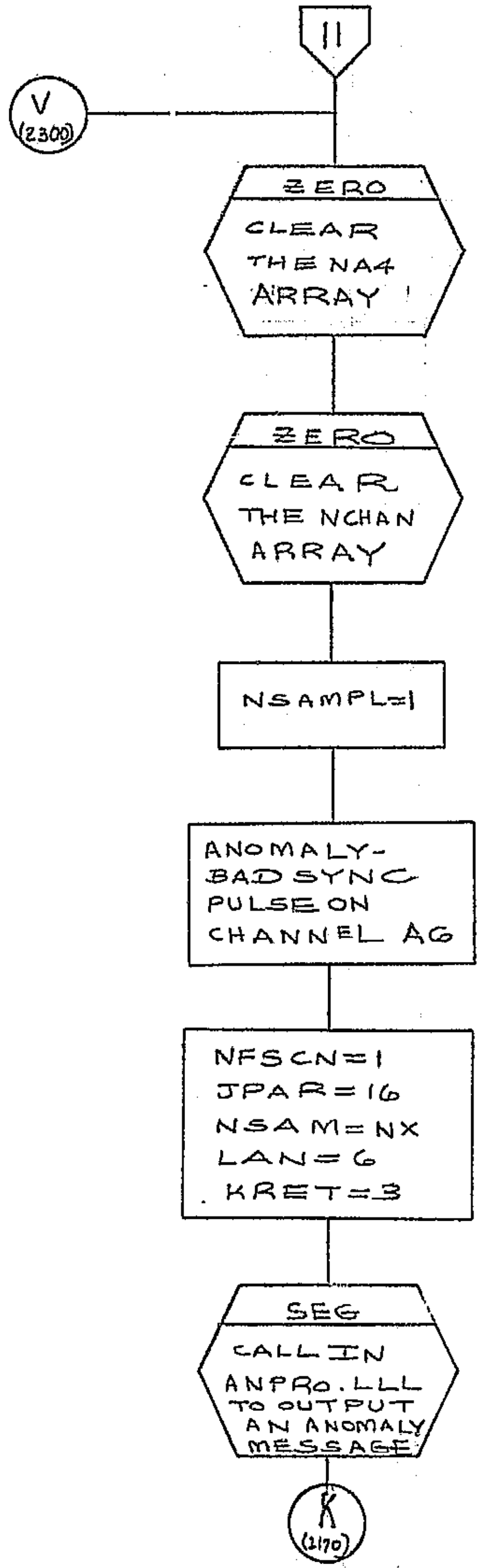

7
MOVE PART OF SCAN TO CH. A4 ARRAY (NA4)
AMOV
TRANSFER DATA FROM THE MEAS ARRAY TO THE NA4 ARRAY
IBACK=1
GO TO
ICHAN=1
ICHAN=3
ICHAN=2
DDDDD (3850)
FFFFF (3959)
EEEEE (3909)
M (2185)
NSAMPL=NSAMPL + NX
NFSCN = 1
NIX = 0
GGGGG (3999)

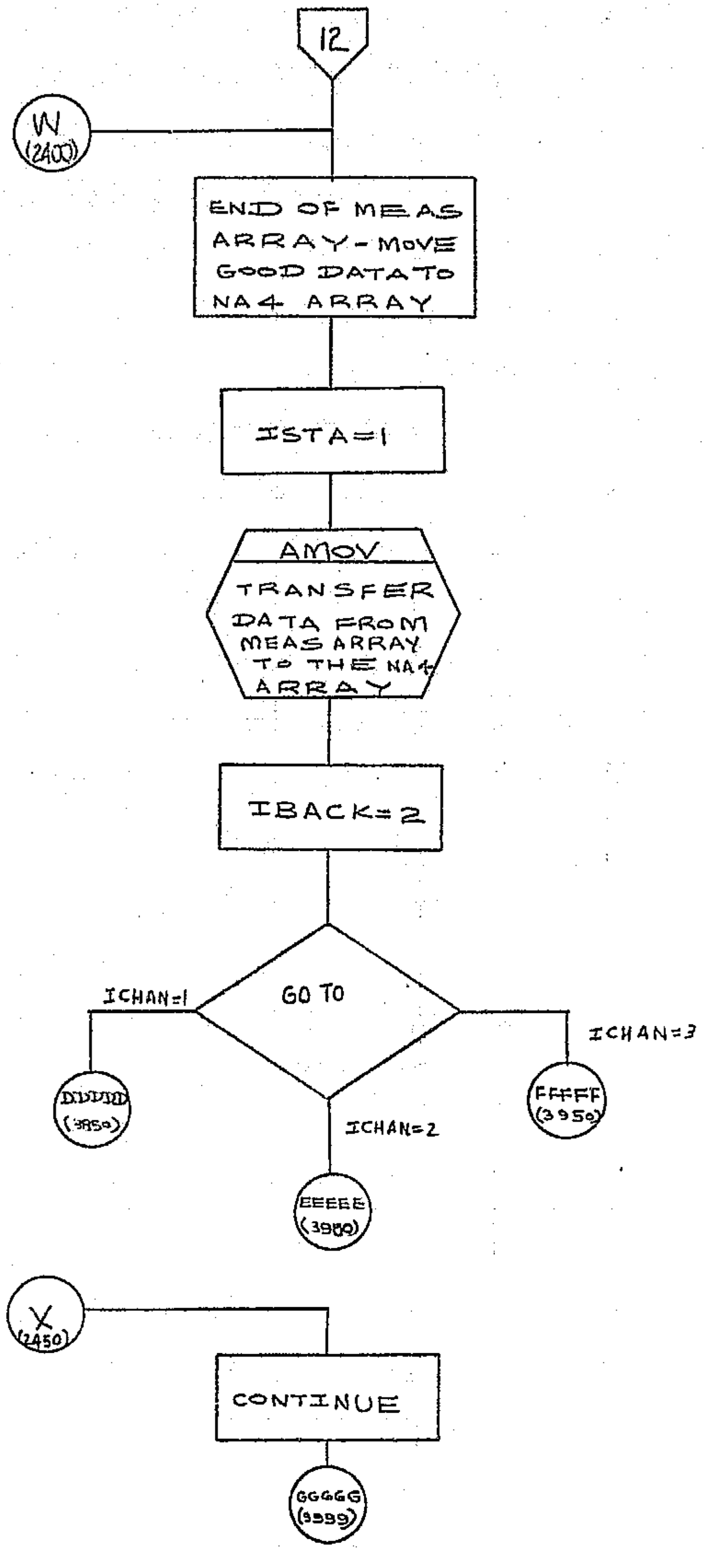

8
N (2190)
CHECK FOR END OF MEAS ARRAY OR DATA PERIOD
NX+1 > IHI ?
Y
W (2400)
N
II = NX+1
P (2210)
O (2200)
II=NX
P (2210)
CHECK NEXT VALUES IN OTHER CHANNELS FOR SYNC PULSES
MEAS (LOCAL+II) < ISYNC ?
Y
Q (2221)
N
9

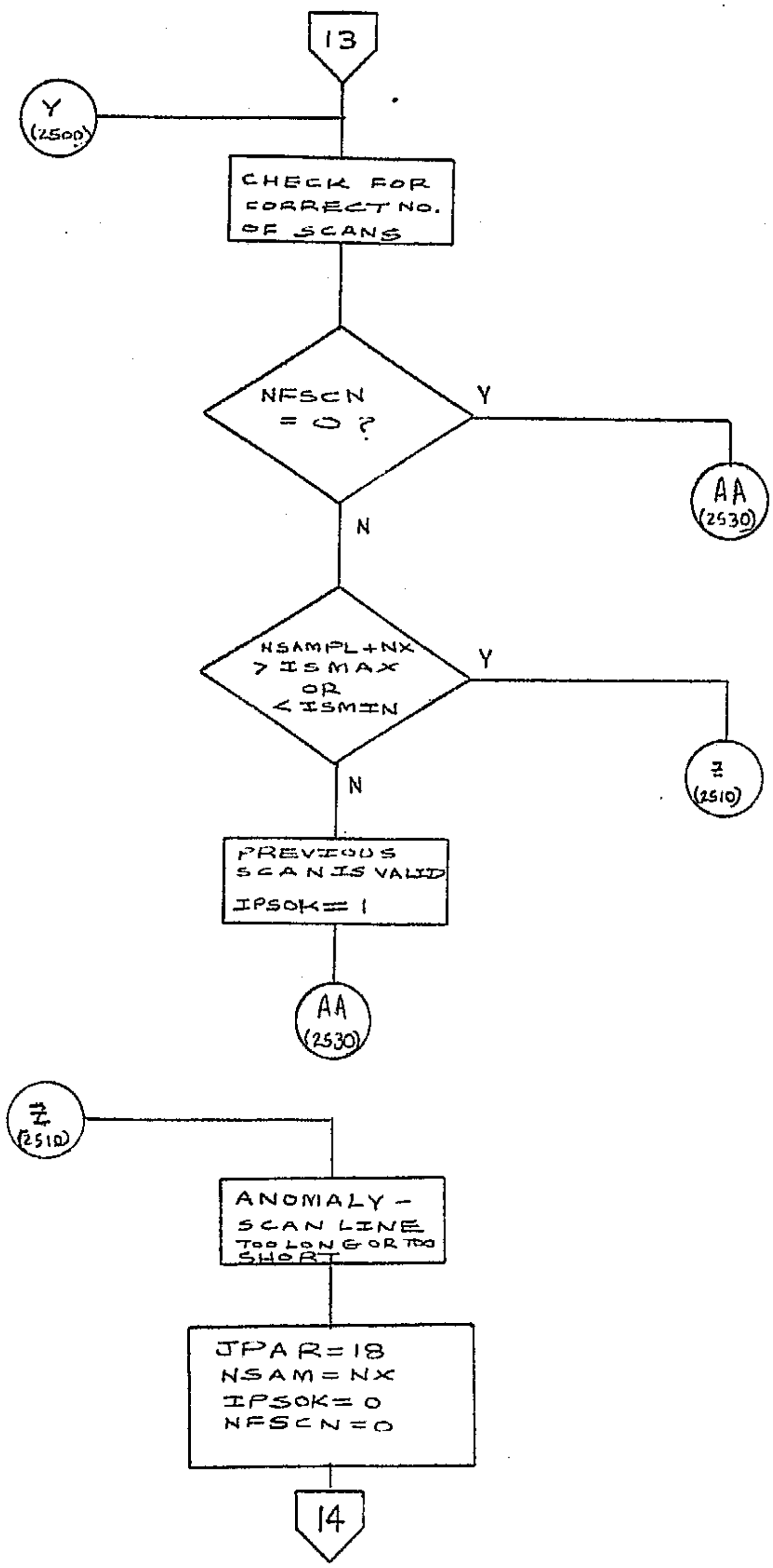

9
MEAS(LOCA2+II) < ISYNC ?
Y
R (2222)
N
MEAS(LOCA3+II) < ISYNC ?
Y
S (2223)
N
MEAS(LOCA5+II) < ISYNC ?
Y
T (2225)
N
Y (2500)
Q (2221)
ANOMALY—BAD SYNC PULSE—SCAN NOT REJECTED
LAN=1
U (2230)

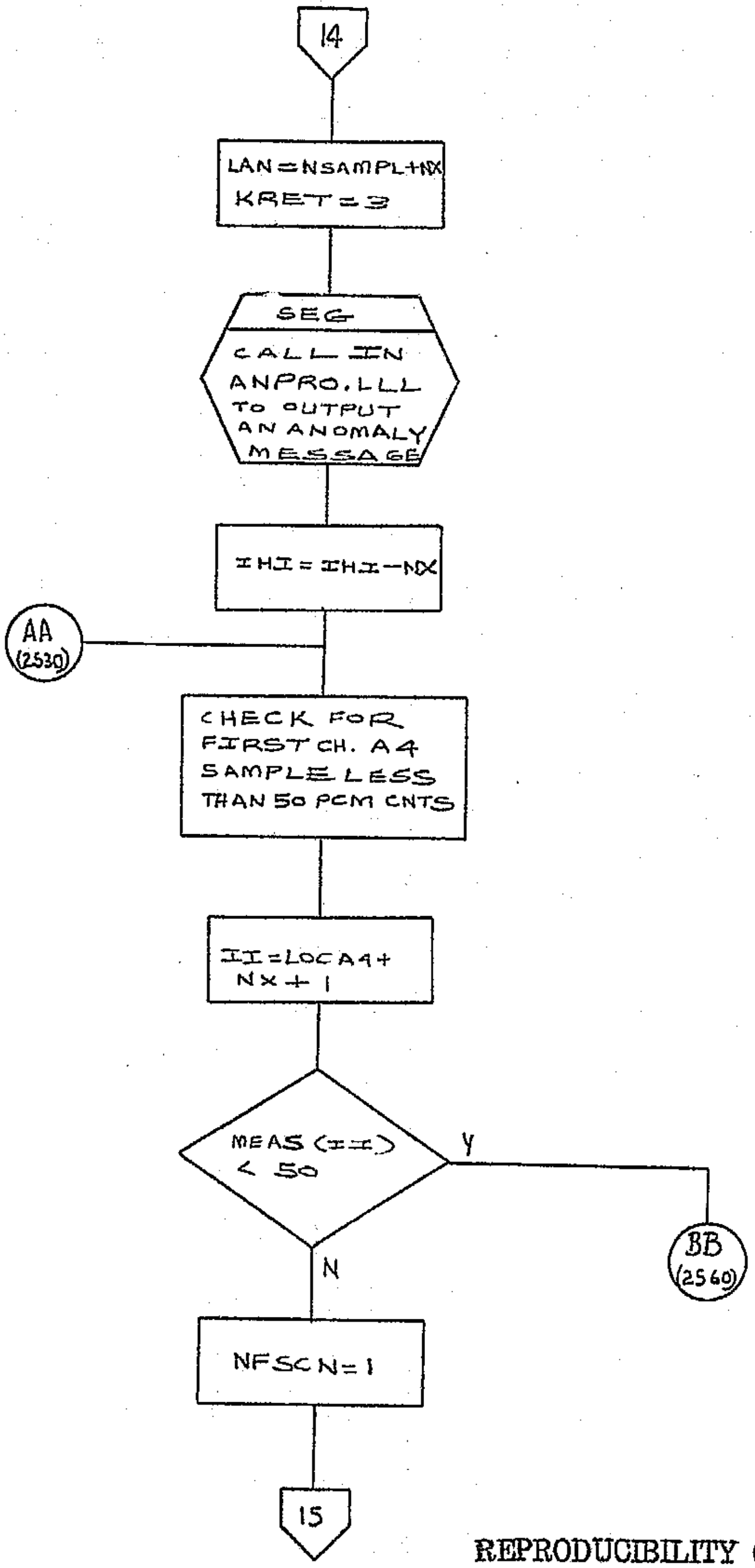


10
R (2222)
LAN = 2
U (2230)
S (2223)
LAN = 3
U (2230)
T (2225)
LAN = 5
U (2230)
JPAR = 16
NSAM = NX
KRET = 3
SEG
CALL IN
ANPRO.LLL
TO OUTPUT
AN ANOMALY
MESSAGE
Y (2509)

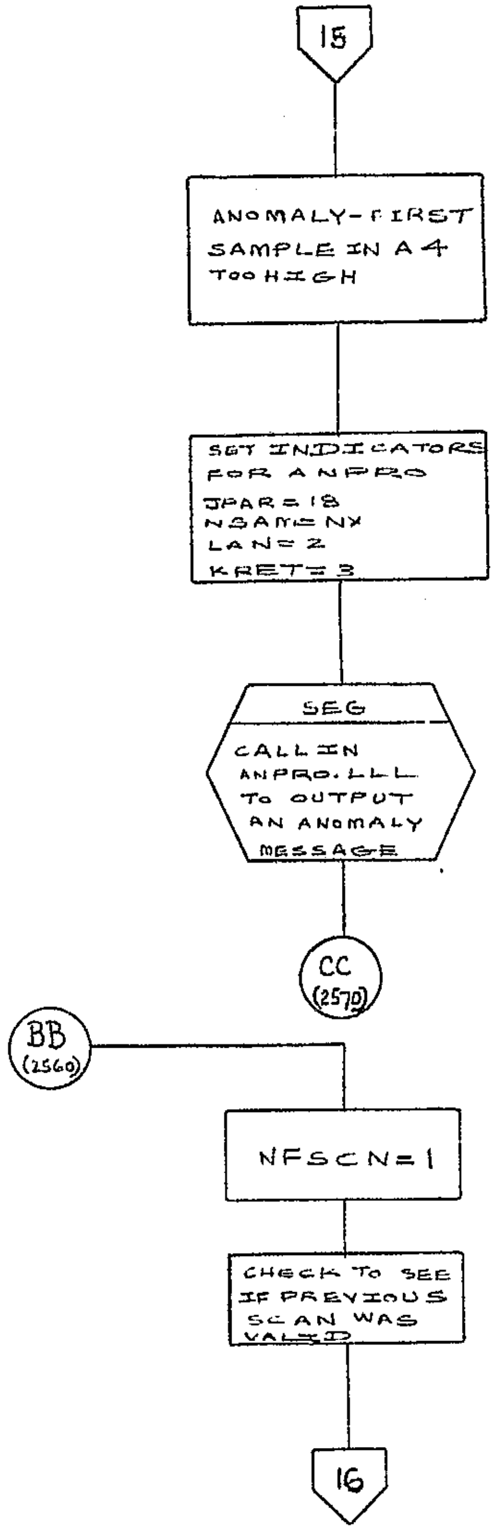

11
V
(2300)
ZERO
CLEAR
THE NA4
ARRAY
ZERO
CLEAR
THE NCHAN
ARRAY
NSAMPL=1
ANOMALY-
BAD SYNC
PULSE ON
CHANNEL AG
NFSCN=1
JPAR=16
NSAM=NX
LAN=6
KRET=3
SEG
CALL IN
ANPRO.LLL
TO OUTPUT
AN ANOMALY
MESSAGE
K
(2170)

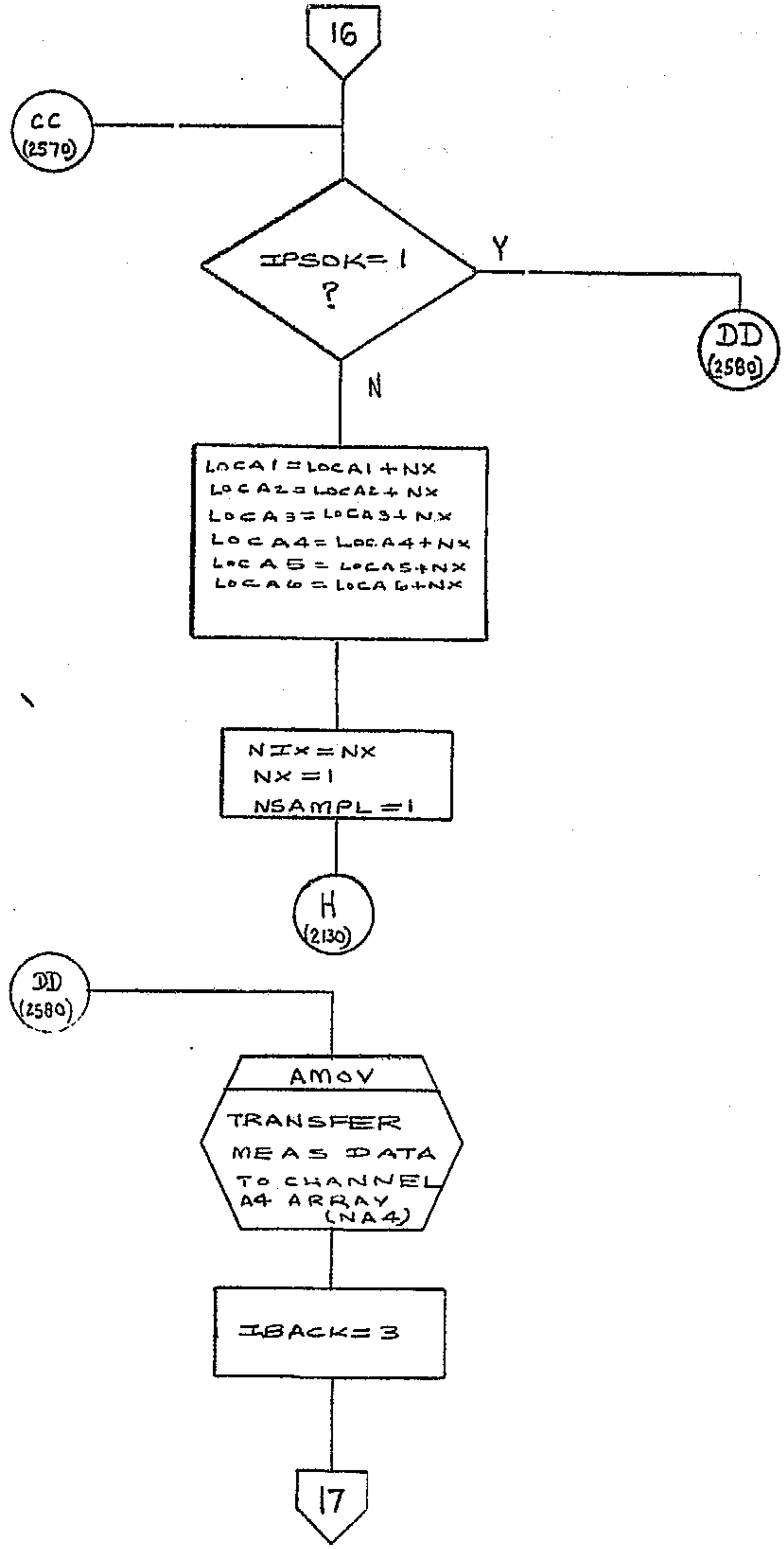

12
W
(2400)
END OF MEAS ARRAY - MOVE GOOD DATA TO NA4 ARRAY
ISTA=1
AMOV
TRANSFER DATA FROM MEAS ARRAY TO THE NA4 ARRAY
IBACK=2
GO TO
ICHAN=1
ICHAN=2
ICHAN=3
DDDDD
(3850)
EEEEE
(3900)
FFFFF
(3950)
X
(2450)
CONTINUE
GGGGG
(3999)

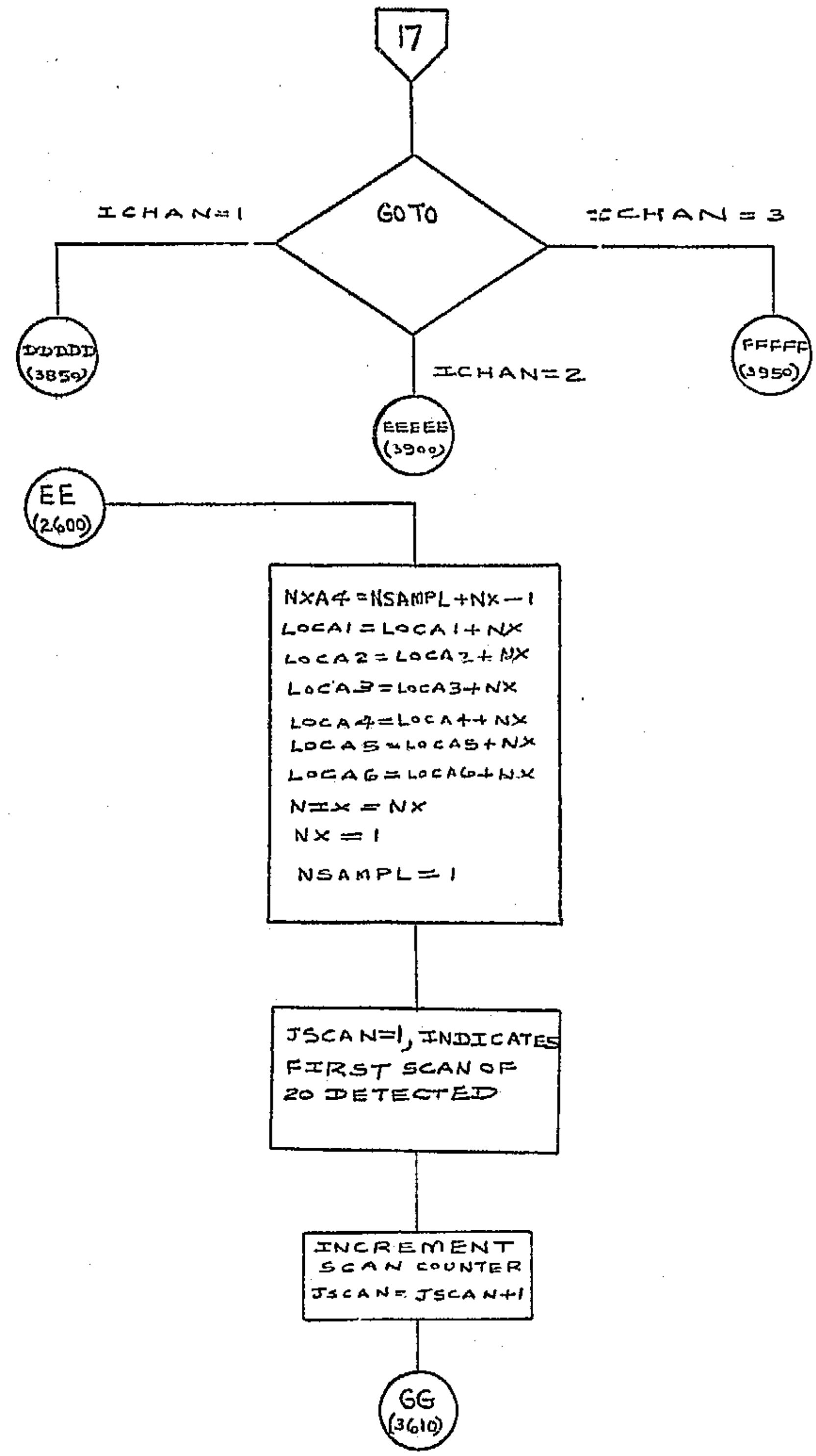

13
Y
(2500)
CHECK FOR CORRECT NO. OF SCANS
NFSCN = 0 ?
Y
AA
(2530)
N
NSAMPL+NX > ISMAX OR < ISMIN
Y
Z
(2510)
N
PREVIOUS SCAN IS VALID
IPSOK = 1
AA
(2530)
Z
(2510)
ANOMALY - SCAN LINE TOO LONG OR TOO SHORT
JPAR=18
NSAM=NX
IPSOK=0
NFSCN=0
14

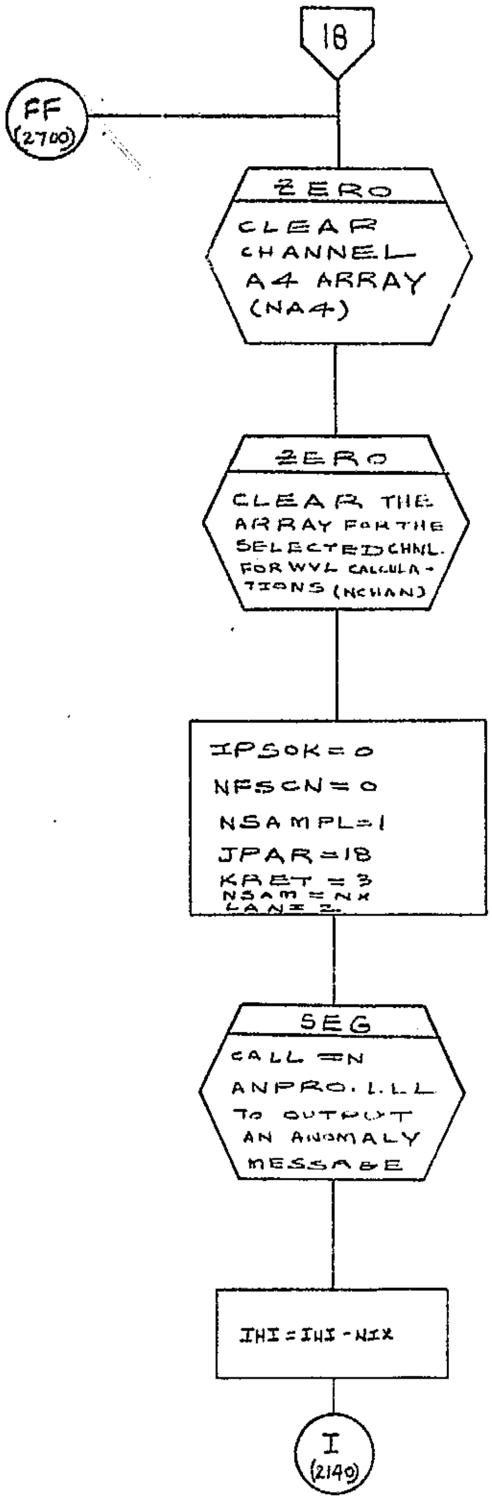


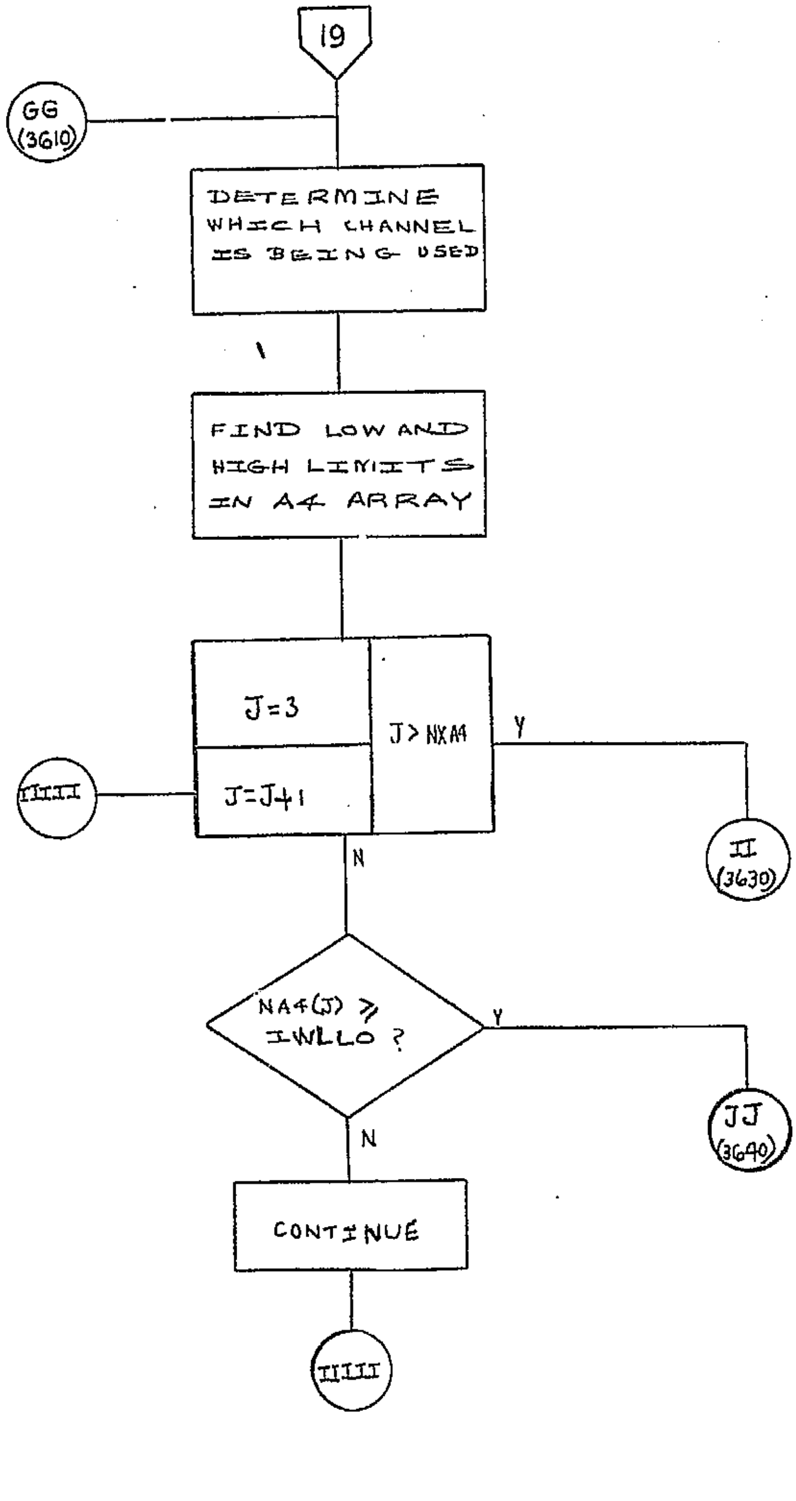
14
LAN=NSAMPL+NX
KRET=3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
IHI=IHI-NX
AA
(2530)
CHECK FOR FIRST CH. A4 SAMPLE LESS THAN 50 PCM CNTS
II=LOCA4+NX+1
MEAS (II) < 50
Y
N
BB
(2560)
NFSCN=1
15

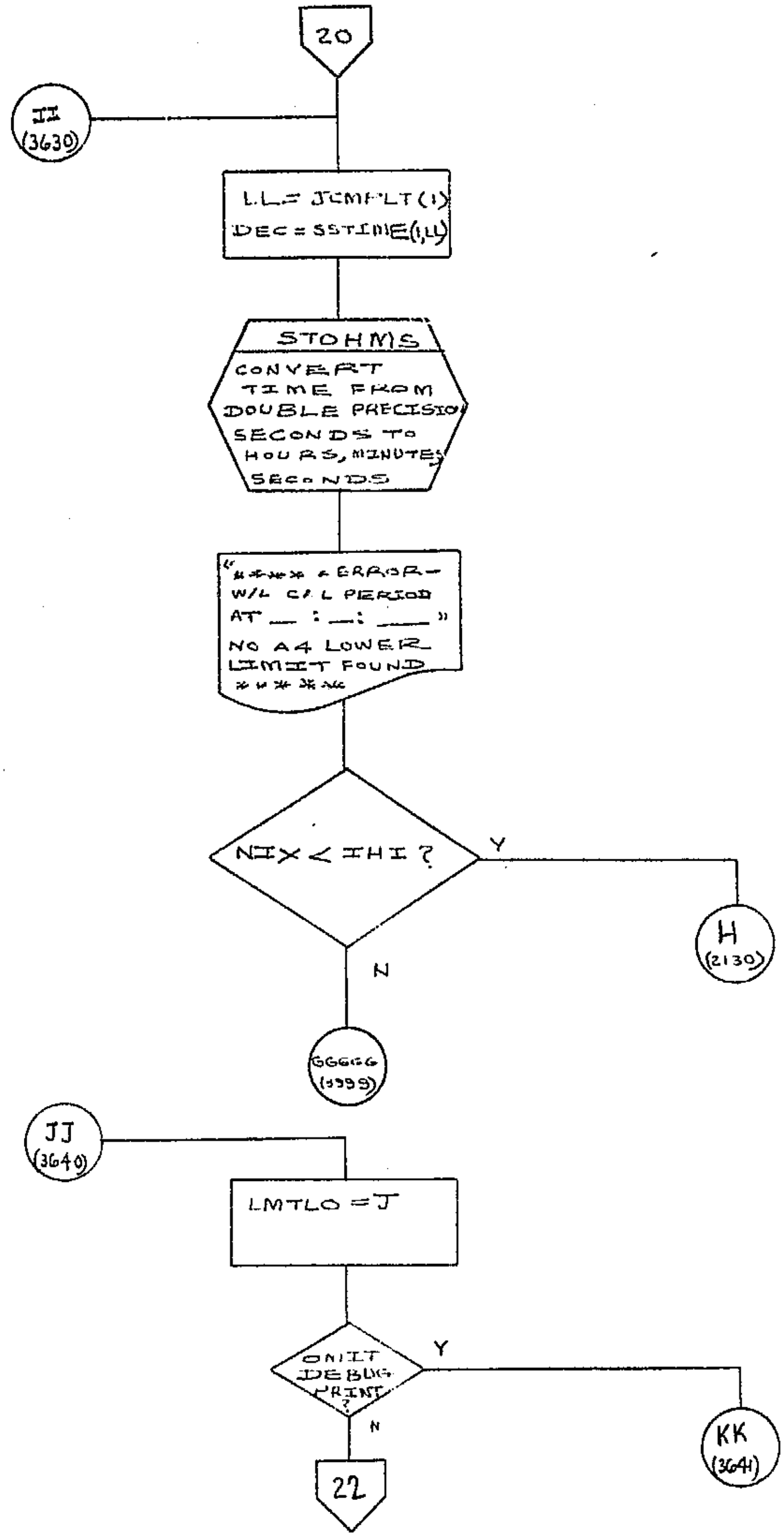

15
ANOMALY-FIRST
SAMPLE IN A4
TOO HIGH
SET INDICATORS
FOR ANPRO
JPAR=18
NSAME=NX
LAN=2
KRET=3
SEG
CALL IN
ANPRO.LLL
TO OUTPUT
AN ANOMALY
MESSAGE
CC
(2570)
BB
(2560)
NFSCN=1
CHECK TO SEE
IF PREVIOUS
SCAN WAS
VALID
16

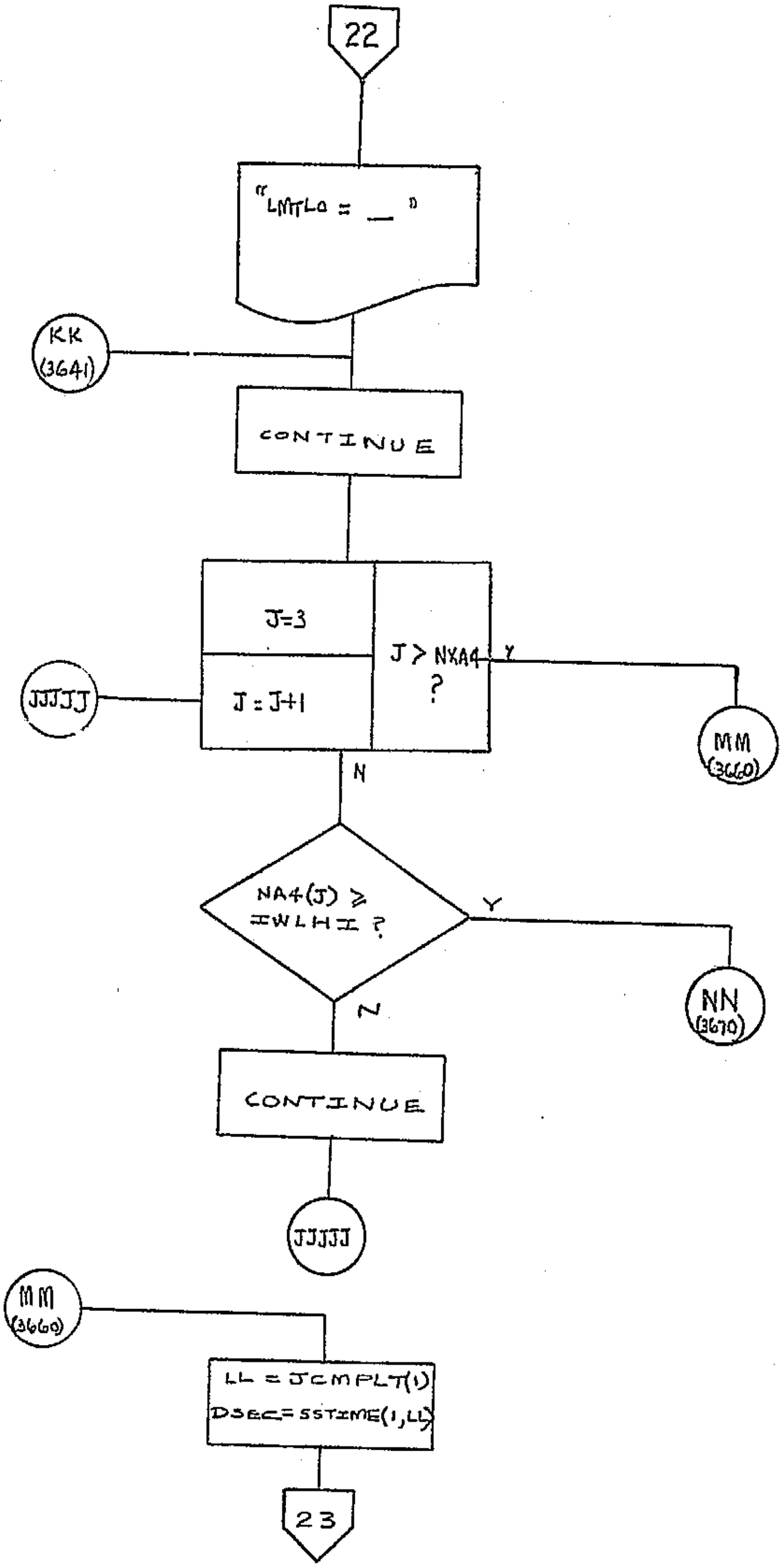

16
CC
(2570)
IPSOK=1
?
Y
DD
(2580)
N
LOCA1=LOCA1+NX
LOCA2=LOCA2+NX
LOCA3=LOCA3+NX
LOCA4=LOCA4+NX
LOCA5=LOCA5+NX
LOCA6=LOCA6+NX
NIX=NX
NX=1
NSAMPL=1
H
(2130)
DD
(2580)
AMOV
TRANSFER
MEAS DATA
TO CHANNEL
A4 ARRAY
(NA4)
IBACK=3
17

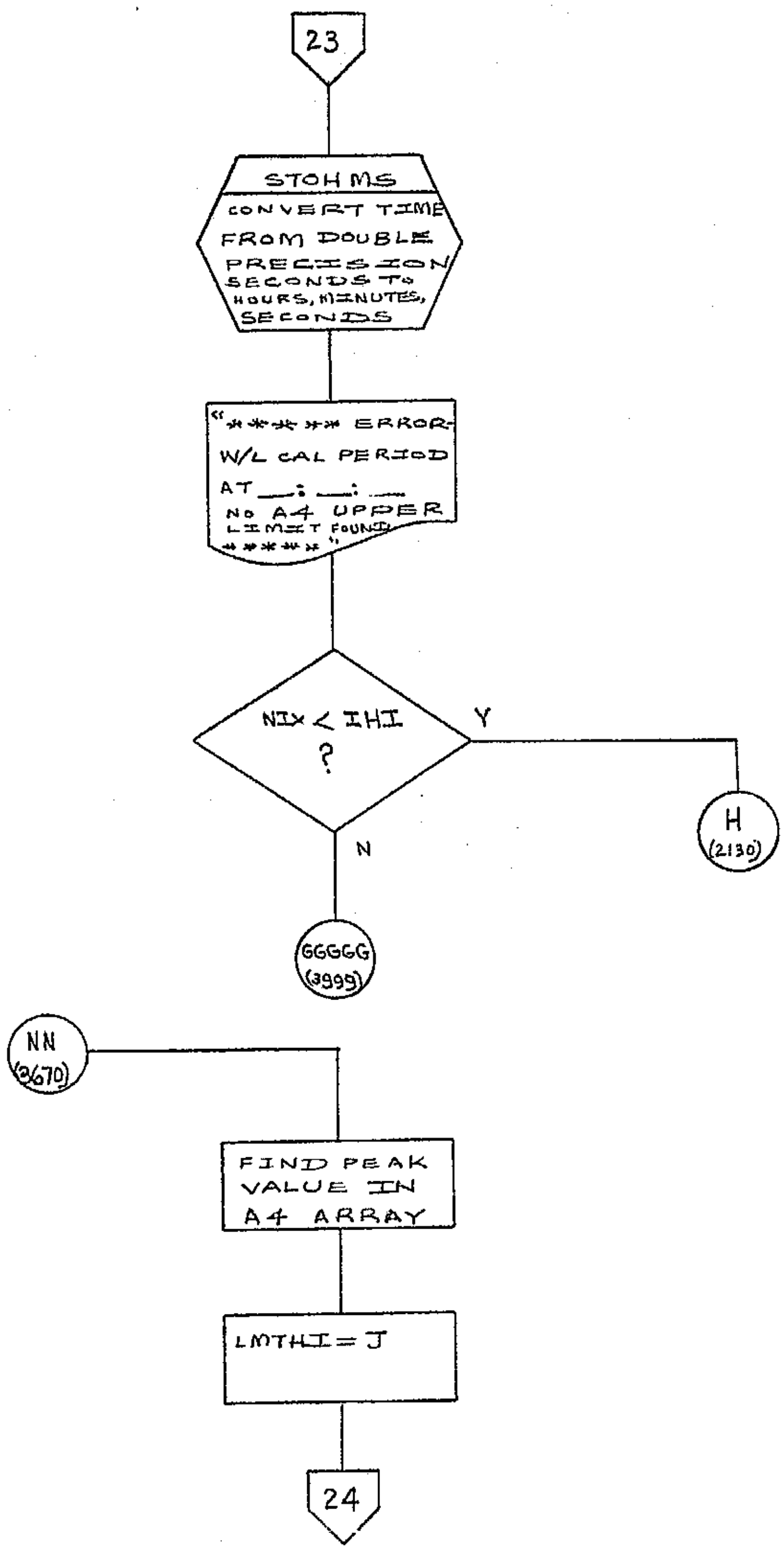

17
GO TO
ICHAN=1
ICHAN=3
ICHAN=2
DDDDD (3850)
EEEEE (3900)
FFFFF (3950)
EE (2600)
NXA4=NSAMPL+NX-1
LOCA1=LOCA1+NX
LOCA2=LOCA2+NX
LOCA3=LOCA3+NX
LOCA4=LOCA4+NX
LOCA5=LOCA5+NX
LOCA6=LOCA6+NX
NIX=NX
NX=1
NSAMPL=1
JSCAN=1, INDICATES FIRST SCAN OF 20 DETECTED
INCREMENT SCAN COUNTER
JSCAN=JSCAN+1
GG (3610)

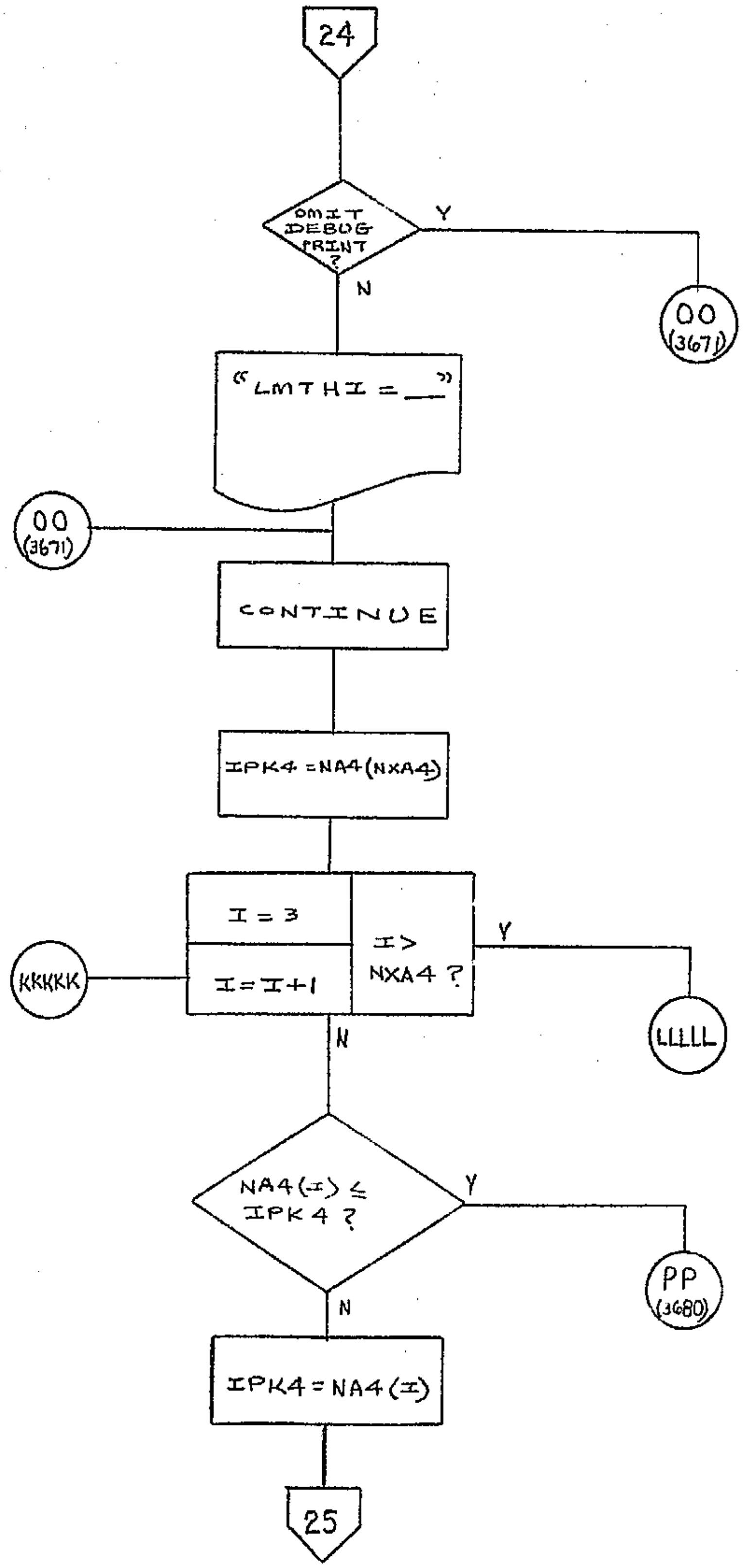

18
FF
(2700)
ZERO
CLEAR CHANNEL A4 ARRAY (NA4)
ZERO
CLEAR THE ARRAY FOR THE SELECTED CHNL. FOR WVL CALCULATIONS (NCHAN)
IPSOK=0
NFSCN=0
NSAMPL=1
JPAR=18
KRET = 3
NSAM = NX
LAN = 2
SEG
CALL ON ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
IHI = IHI - NIX
I
(2140)

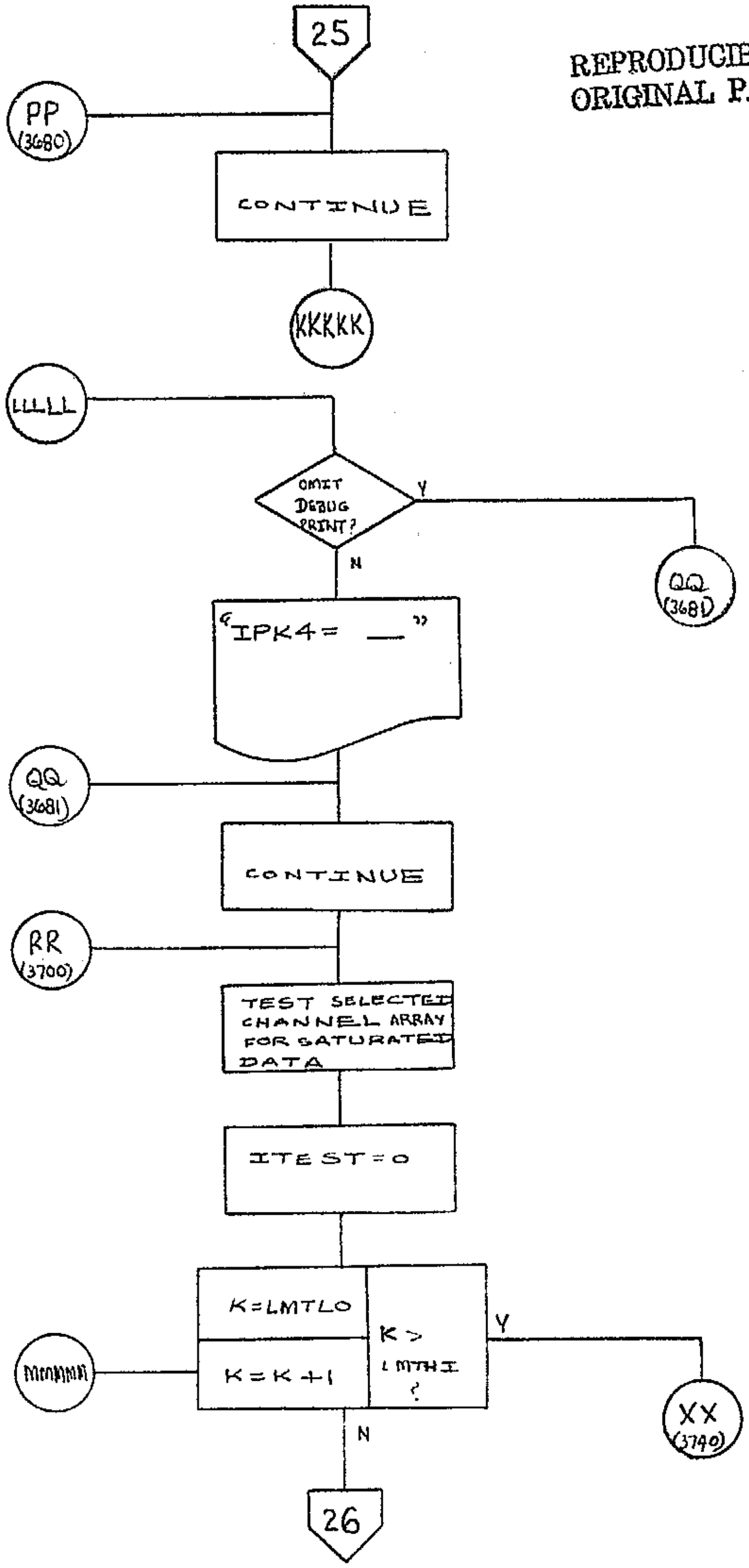

19
GG
(3610)
DETERMINE WHICH CHANNEL IS BEING USED
FIND LOW AND HIGH LIMITS IN A4 ARRAY
J=3
J=J+1
J > NXA4
Y
N
IIIII
II
(3630)
NA4(J) ≥ IWLLO ?
Y
N
JJ
(3640)
CONTINUE
IIIII

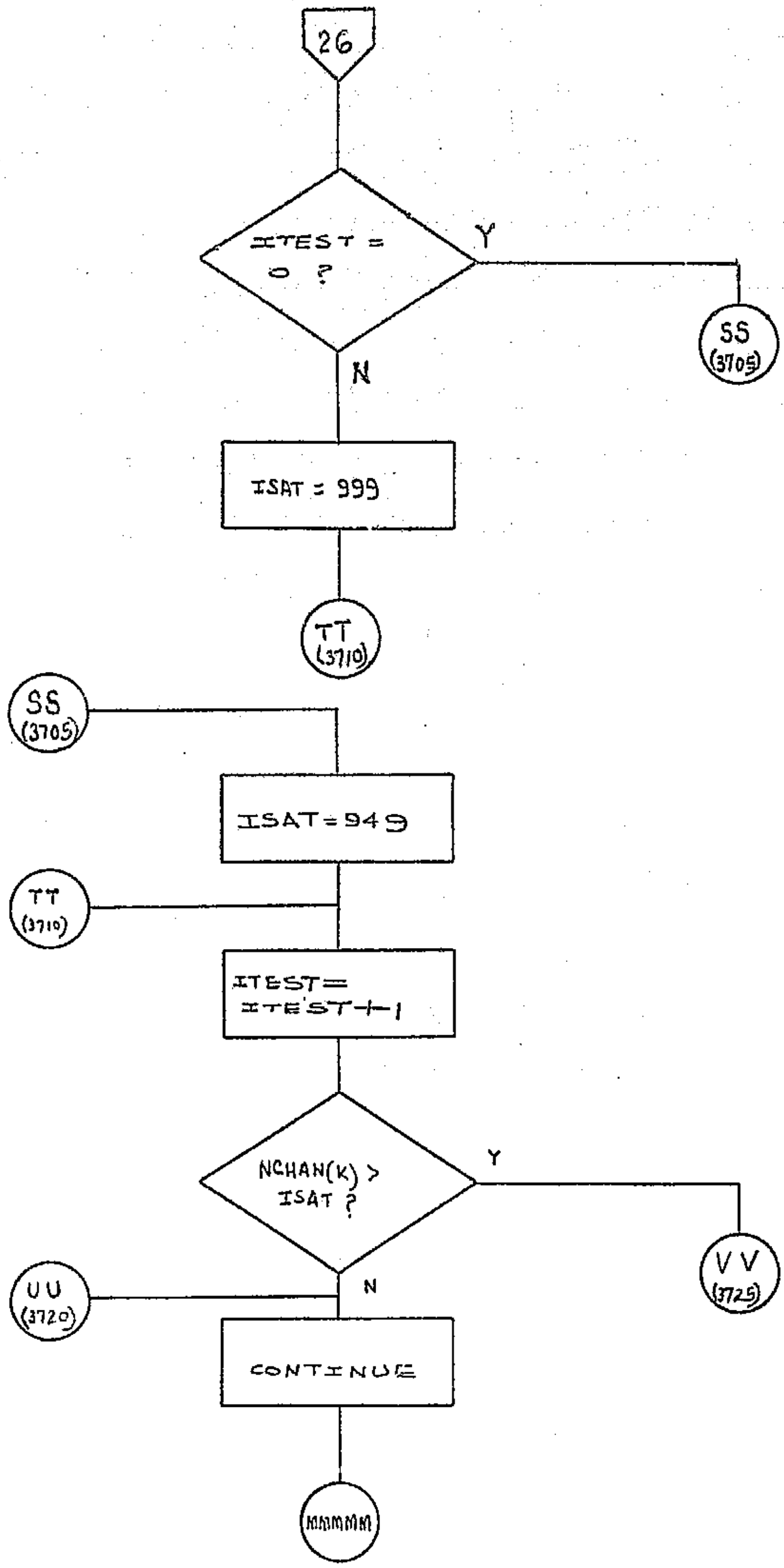

20
II
(3630)
LL = JCMPLT(1)
DEC = SSTIME(1,LL)
STOHMS
CONVERT TIME FROM DOUBLE PRECISION SECONDS TO HOURS, MINUTES, SECONDS
"**** ERROR- W/L CAL PERIOD AT __ : __ : ____ "
NO A4 LOWER LIMIT FOUND *****
NIX < IHI ?
Y
H
(2130)
N
GGGGG
(9999)
JJ
(3640)
LMTLO = J
ONIIT DEBUG PRINT ?
Y
N
KK
(3641)
22

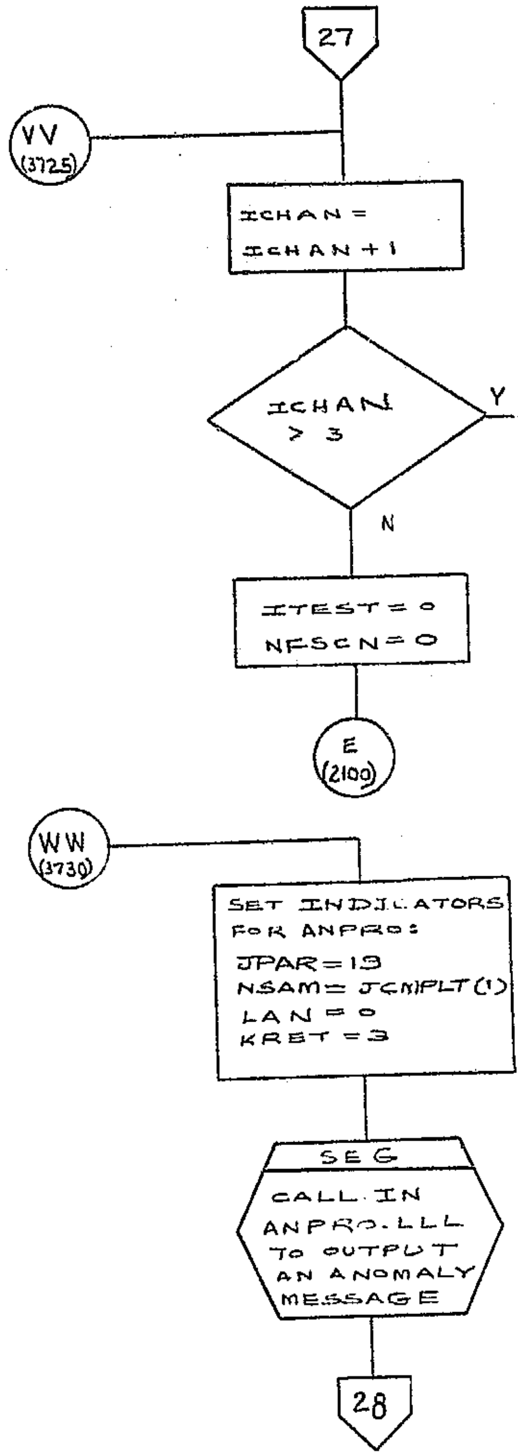

22
"LMTLO = __ "
KK (3641)
CONTINUE
J=3
J = J+1
J > NXA4 ?
Y
N
JJJJJ
MM (3660)
NA4(J) ≥ IWLHI ?
Y
N
NN (3670)
CONTINUE
JJJJJ
MM (3660)
LL = JCMPLT(1)
DSEC = SSTIME(1,LL)
23

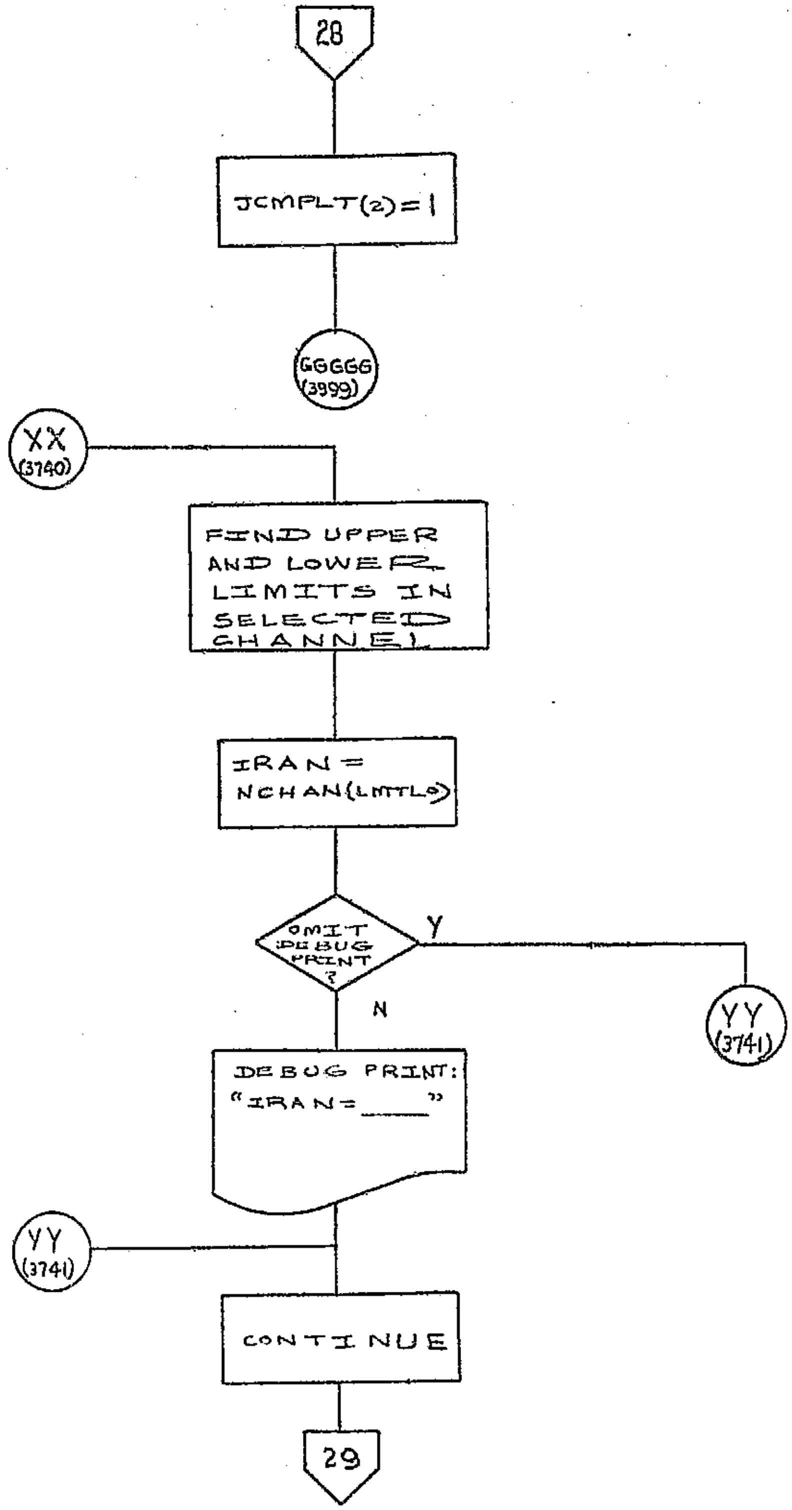

23
STOH MS
CONVERT TIME FROM DOUBLE PRECISION SECONDS TO HOURS, MINUTES, SECONDS
"****** ERROR-W/L CAL PERIOD AT __:__:__ NO A4 UPPER LIMIT FOUND *****"
NIX < IHI ?
Y
H (2130)
N
GGGGG (3999)
NN (2670)
FIND PEAK VALUE IN A4 ARRAY
LMTHI = J
24

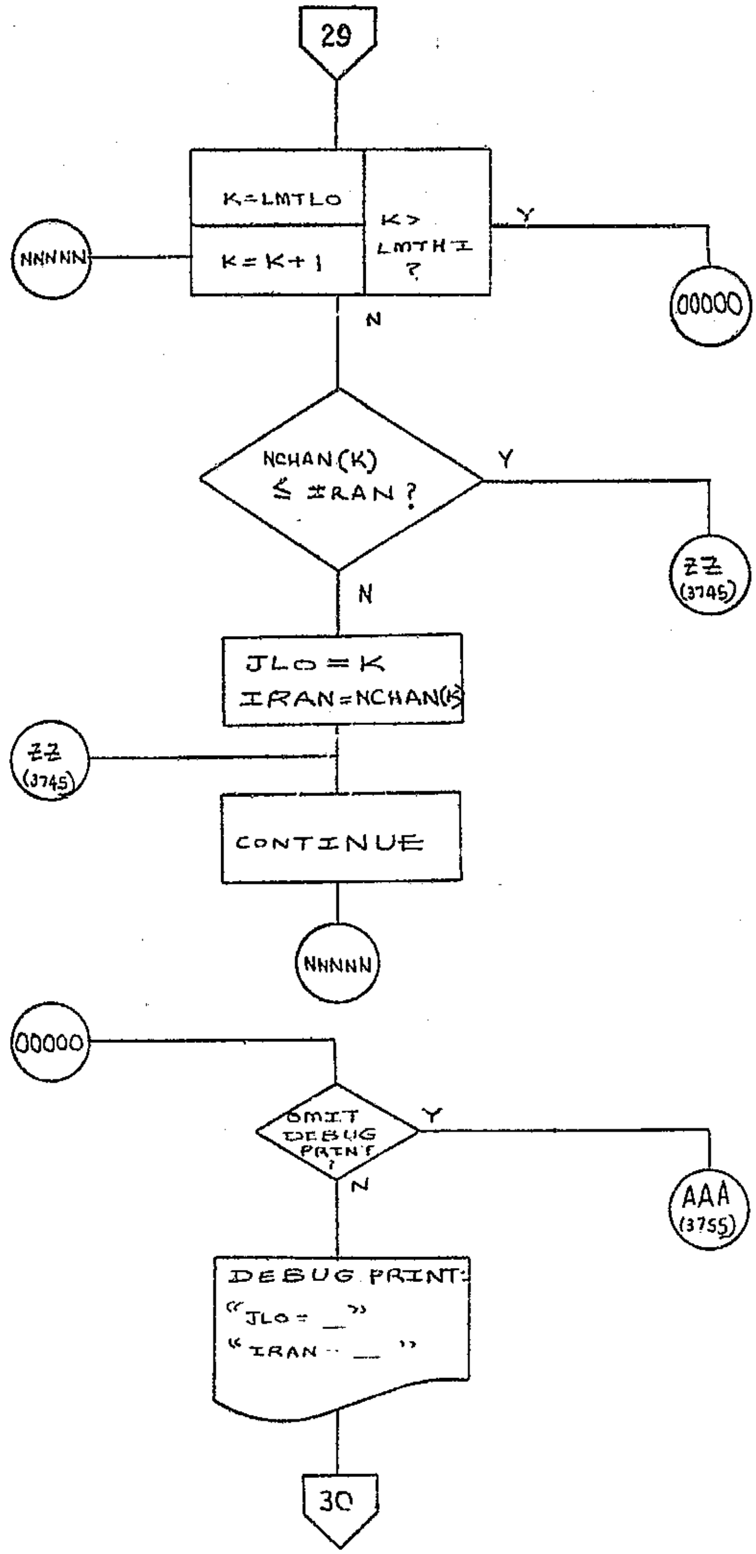

24
OMIT DEBUG PRINT ?
Y
N
OO (3671)
"LMTHI = __"
OO (3671)
CONTINUE
IPK4 = NA4(NXA4)
I = 3
I = I+1
I > NXA4 ?
KKKKK
Y
LLLLL
N
NA4(I) ≤ IPK4 ?
Y
PP (3680)
N
IPK4 = NA4(I)
25

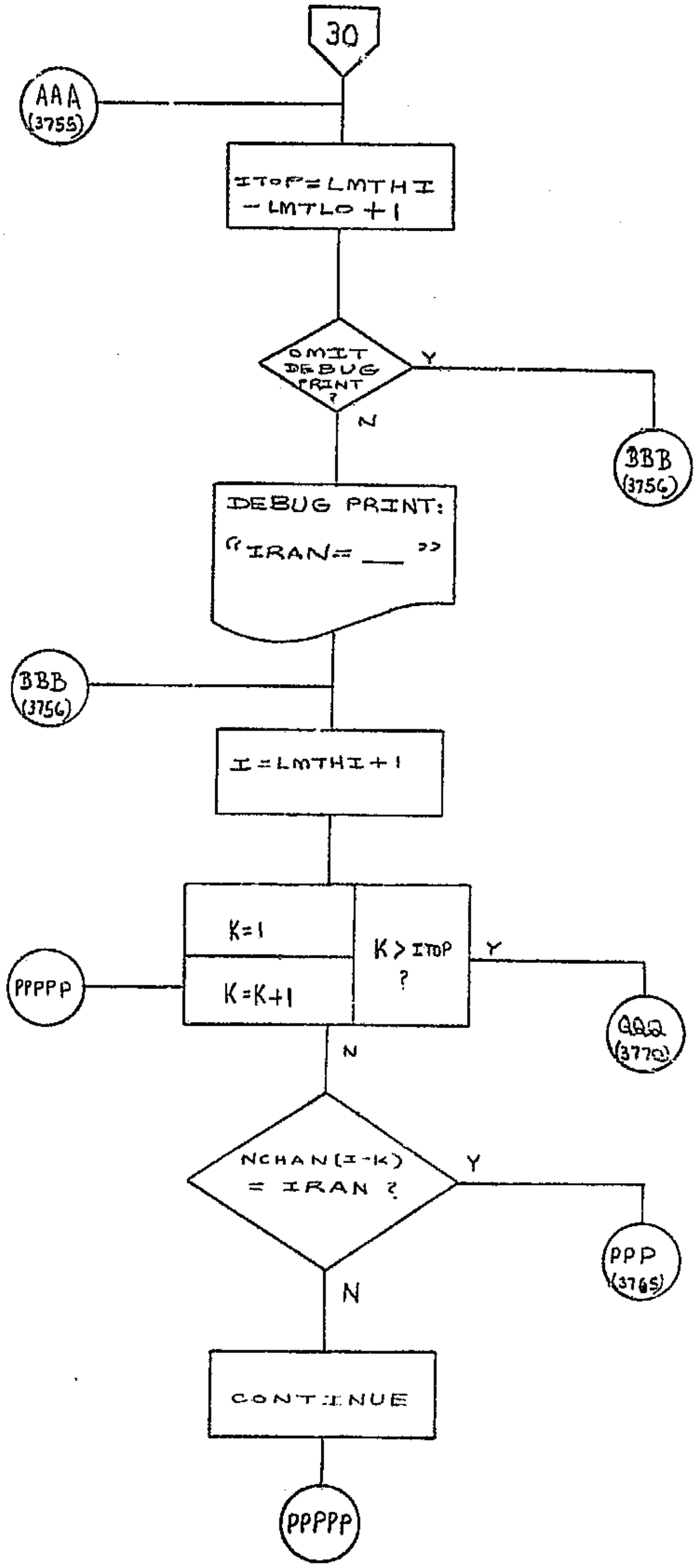

25
PP (3680)
CONTINUE
KKKKK
LLLLL
OMIT DEBUG PRINT?
Y
N
QQ (3681)
"IPK4= __"
QQ (3681)
CONTINUE
RR (3700)
TEST SELECTED CHANNEL ARRAY FOR SATURATED DATA
ITEST=0
K=LMTLO
K=K+1
K > LMTHI ?
Y
N
MMMMM
XX (3740)
26

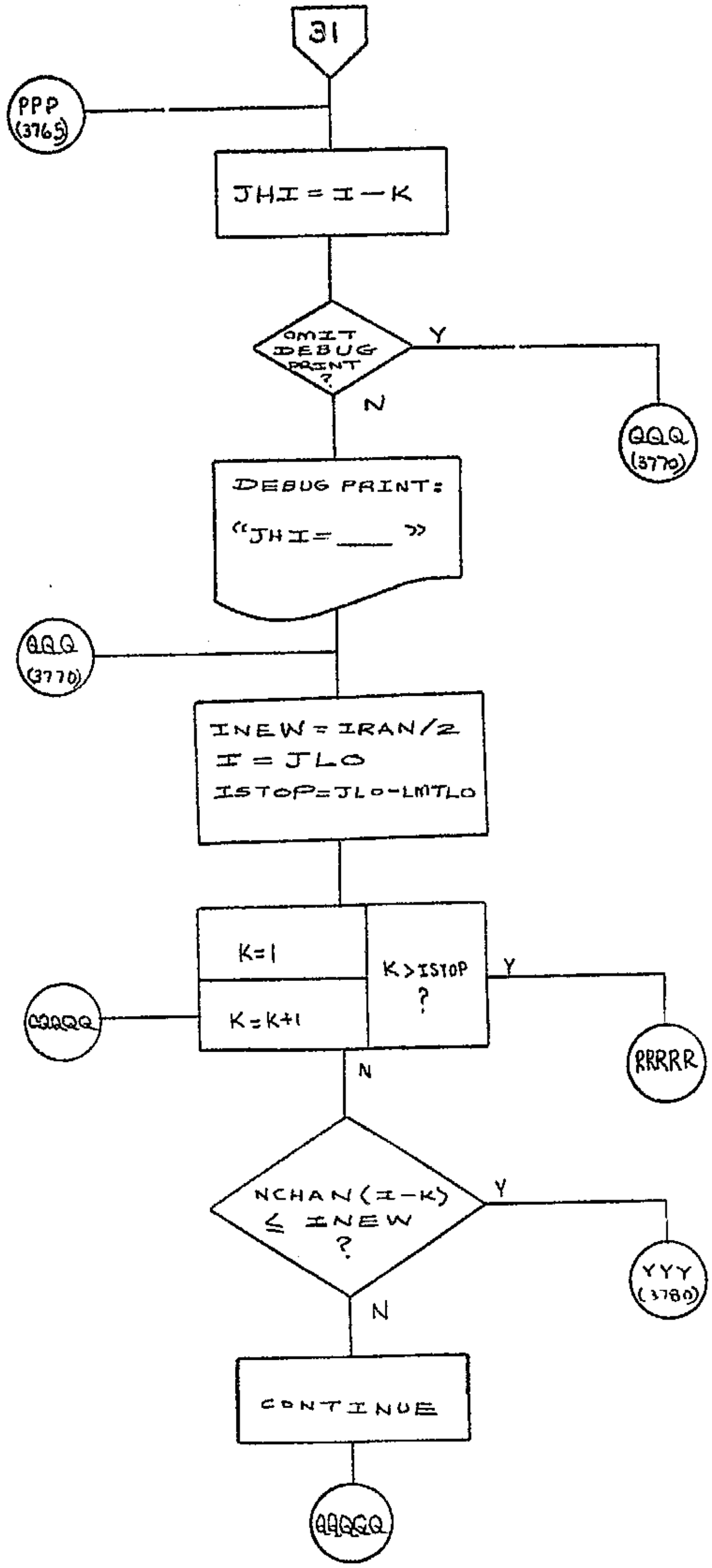

26
ITEST = 0 ?
Y
N
SS (3705)
ISAT = 999
TT (3710)
SS (3705)
ISAT=949
TT (3710)
ITEST= ITEST+1
NCHAN(K) > ISAT ?
Y
N
VV (3725)
UU (3720)
CONTINUE
MMMMM

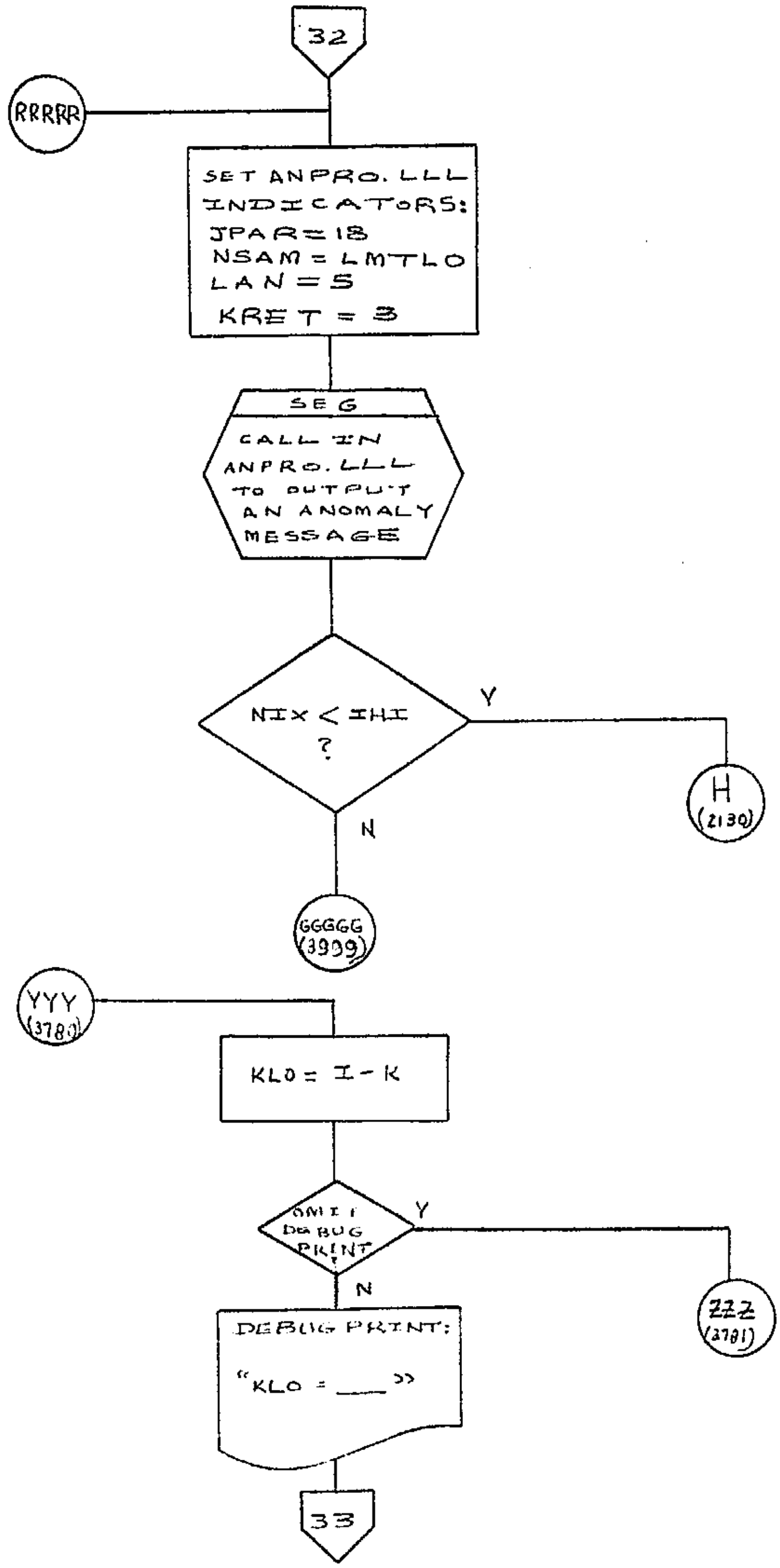

27
VV (3725)
ICHAN = ICHAN + 1
ICHAN > 3
Y
WW (3730)
N
ITEST = 0
NFSCN = 0
E (2100)
WW (3730)
SET INDICATORS FOR ANPRO:
JPAR = 13
NSAM = JCMPLT(1)
LAN = 0
KRET = 3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
28

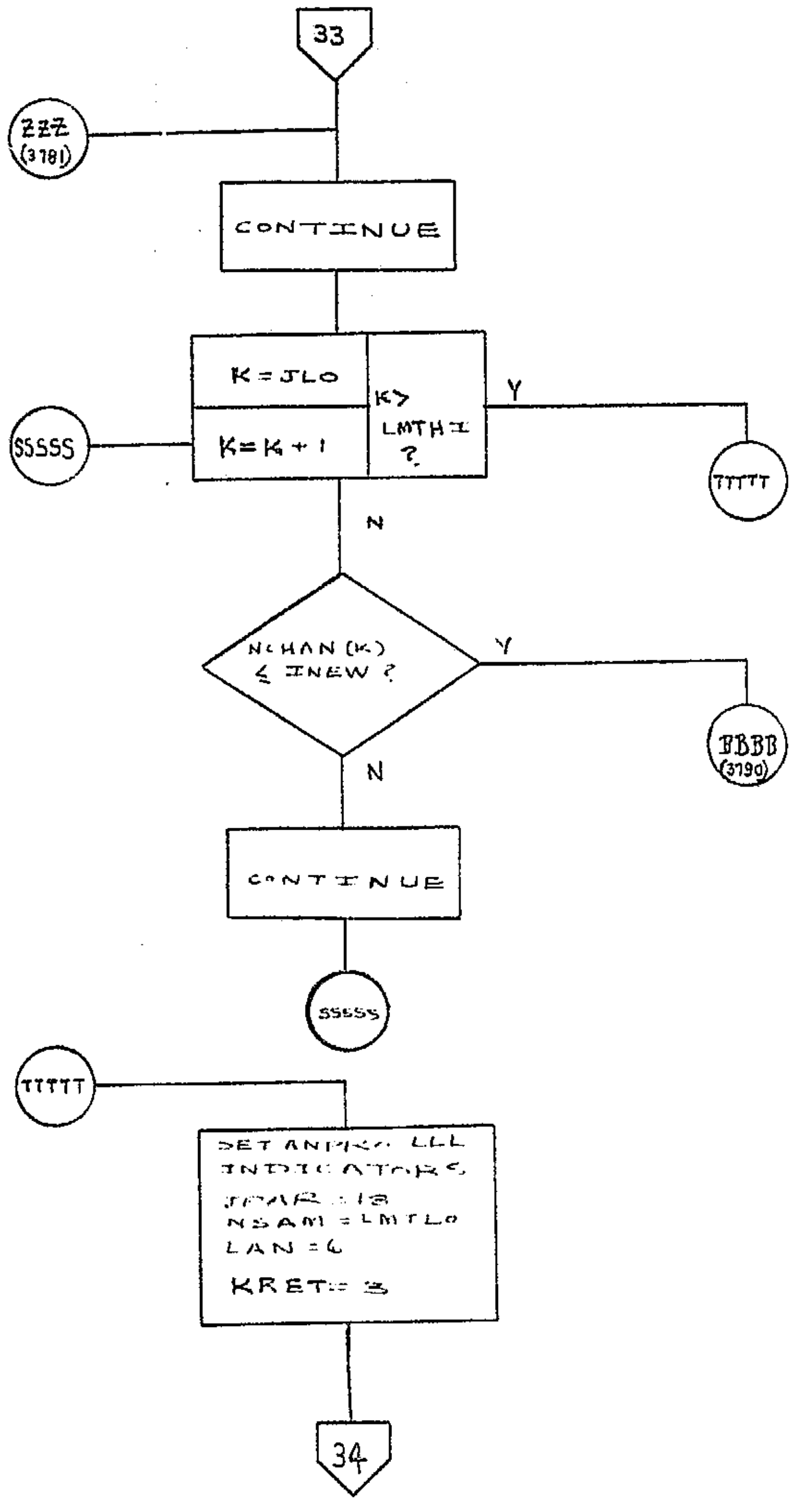

28
JCMPLT(2)=1
GGGGG (3999)
XX (3740)
FIND UPPER AND LOWER LIMITS IN SELECTED CHANNEL
IRAN = NCHAN(LMTLO)
OMIT DEBUG PRINT ?
Y
N
YY (3741)
DEBUG PRINT: "IRAN=____"
YY (3741)
CONTINUE
29

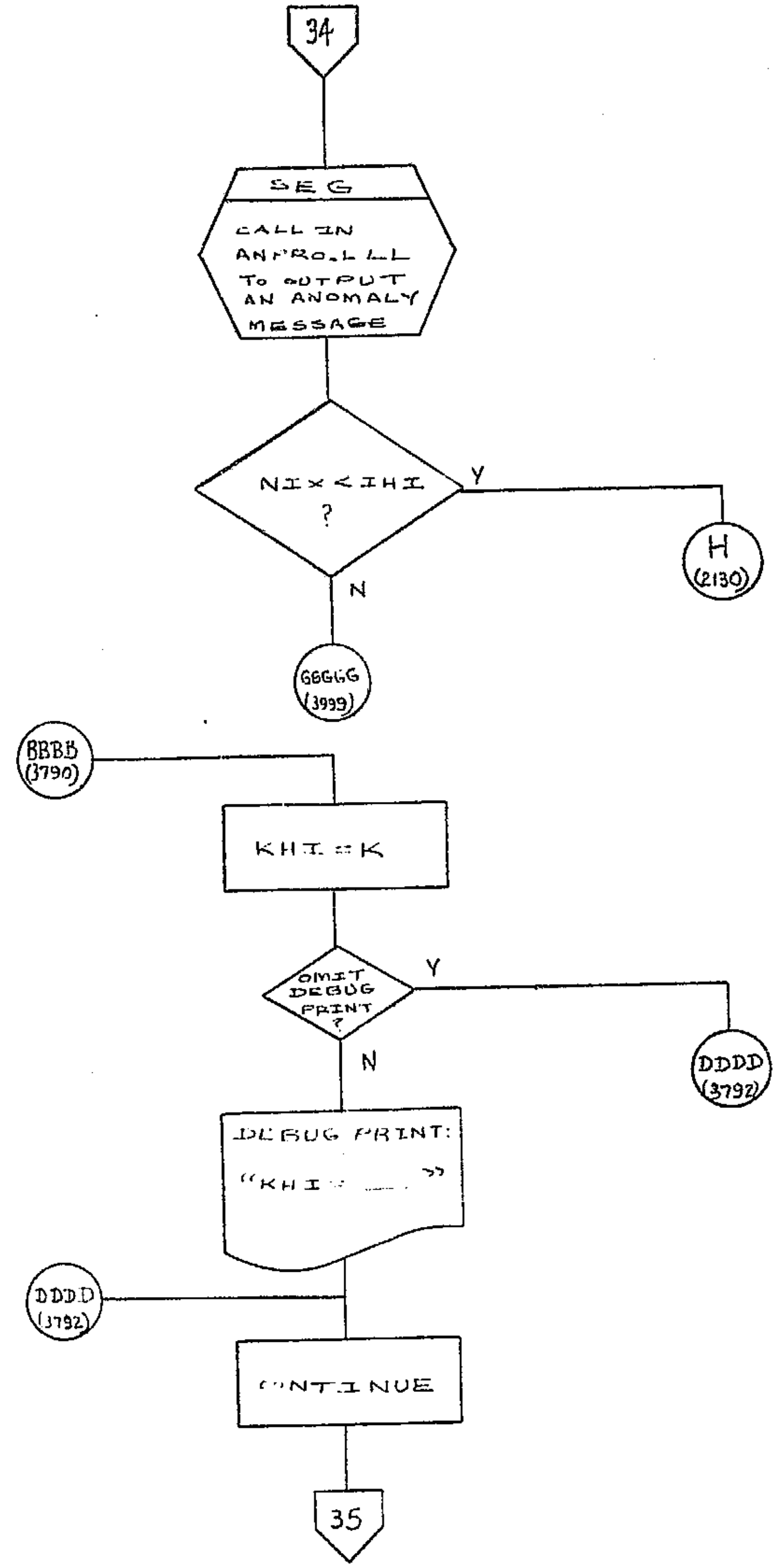

29
K=LMTLO
K=K+1
K>
LMTHI
?
Y
N
NNNNN
00000
NCHAN(K)
≤ IRAN?
Y
N
ZZ
(3745)
JLO=K
IRAN=NCHAN(K)
ZZ
(3745)
CONTINUE
NNNNN
00000
OMIT
DEBUG
PRINT
?
Y
N
AAA
(3755)
DEBUG PRINT:
"JLO= __"
"IRAN= __"
30

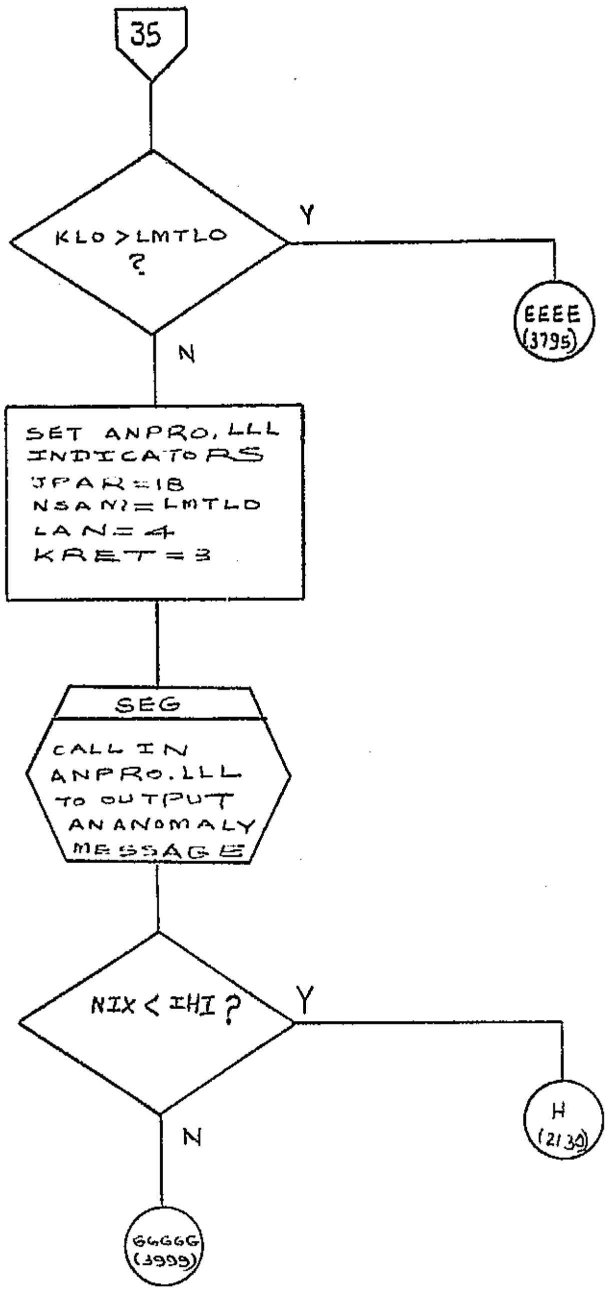

30
AAA (3755)
ITOP=LMTHI -LMTLO+1
OMIT DEBUG PRINT ?
Y
N
BBB (3756)
DEBUG PRINT:
"IRAN= ___ "
BBB (3756)
I=LMTHI+1
K=1
K=K+1
K > ITOP ?
Y
N
PPPPP
QQQ (3770)
NCHAN(I-K) = IRAN ?
Y
N
PPP (3765)
CONTINUE
PPPPP

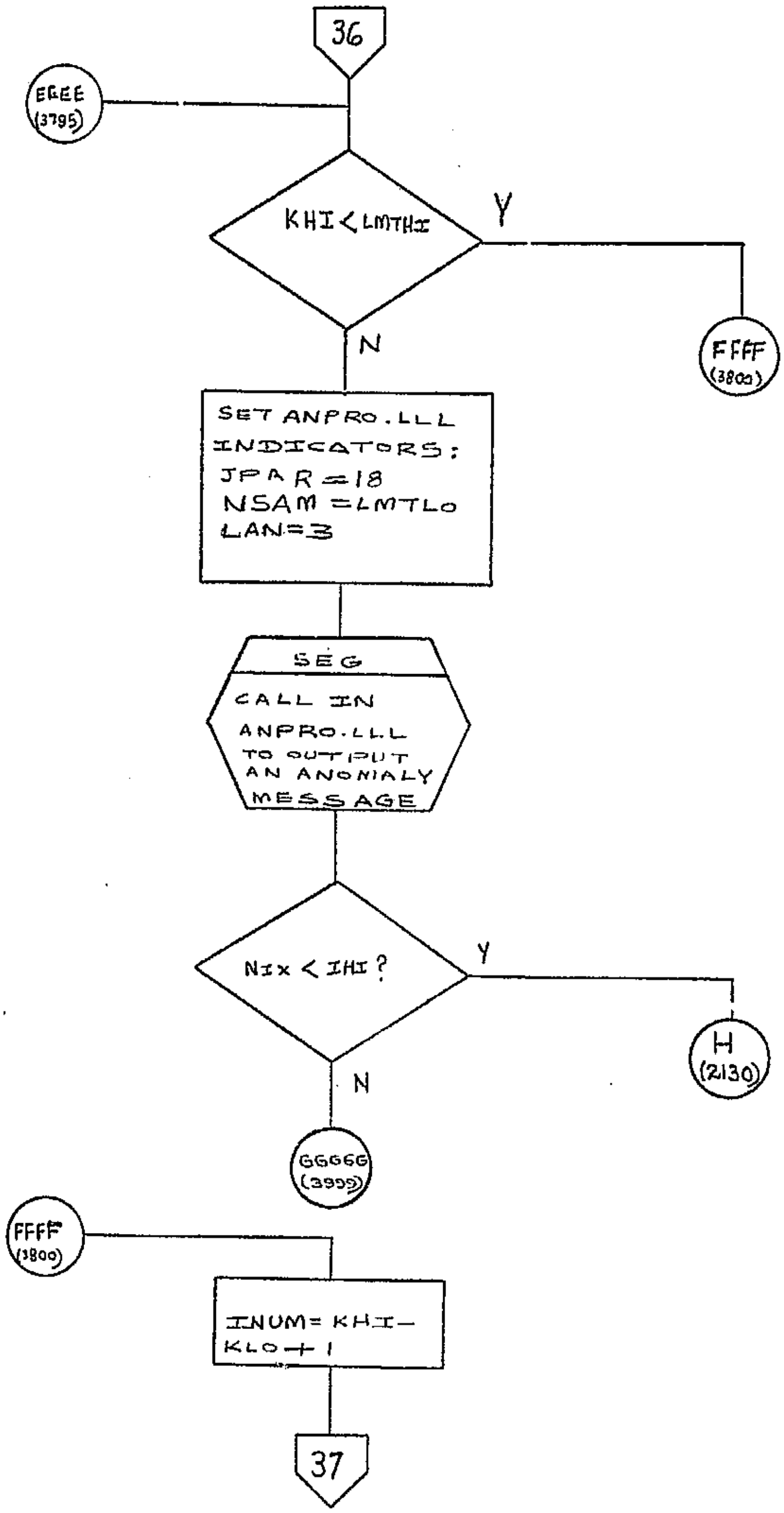

31
PPP (3765)
JHI = I — K
OMIT DEBUG PRINT ?
Y
N
QQQ (3770)
DEBUG PRINT:
"JHI=____"
QQQ (3770)
INEW = IRAN/2
I = JLO
ISTOP=JLO-LMTLO
K=1
K=K+1
K>ISTOP ?
Y
QQQQQ
RRRRR
N
NCHAN (I—K) ≤ INEW ?
Y
YYY (3780)
N
CONTINUE
QQQQQ

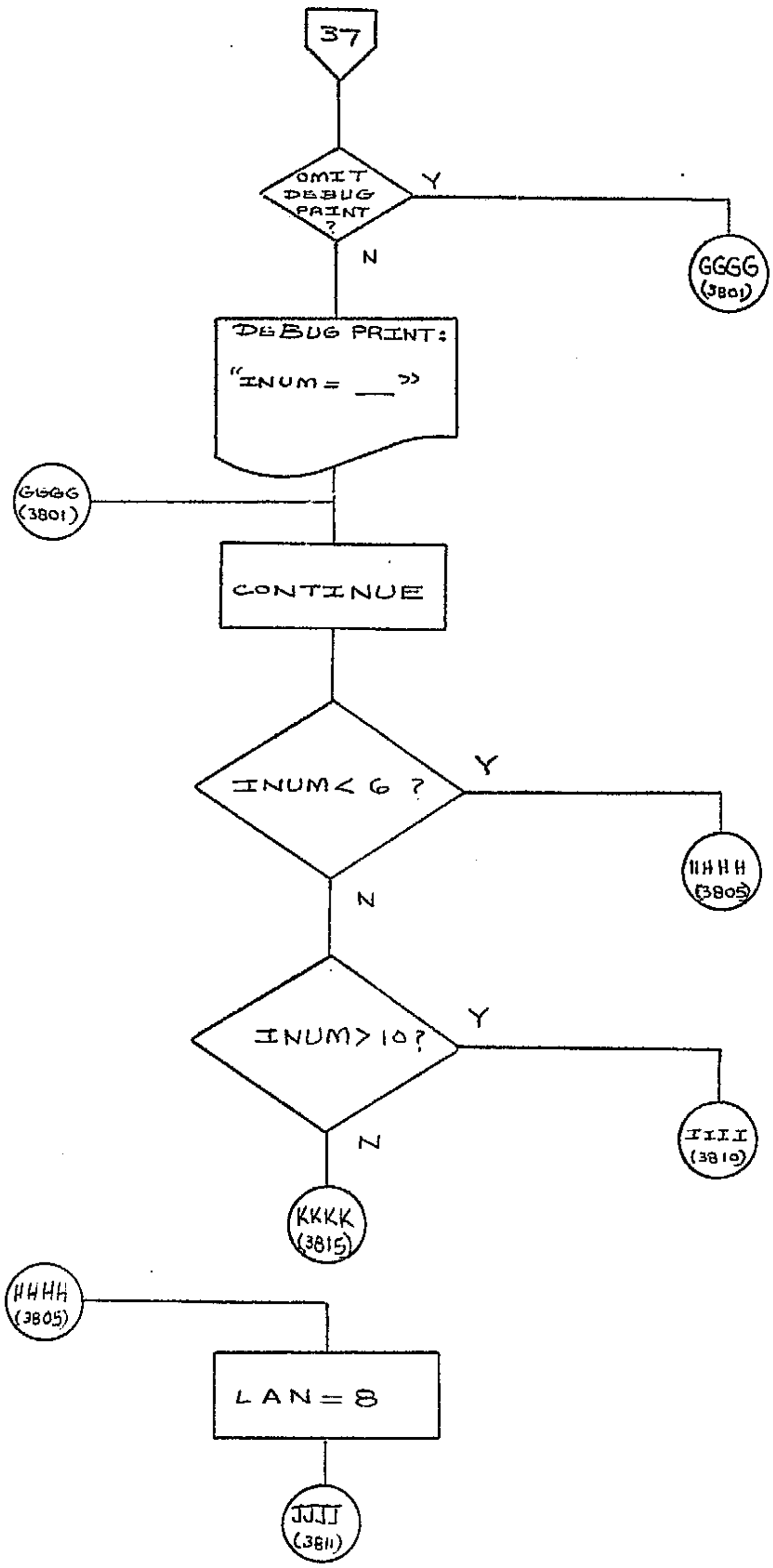

32
RRRRR
SET ANPRO.LLL INDICATORS:
JPAR=18
NSAM = LMTLO
LAN=5
KRET = 3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
NIX < IHI ?
Y
N
H (2130)
GGGGG (3999)
YYY (3780)
KLO = I-K
OMIT DEBUG PRINT
Y
N
ZZZ (3781)
DEBUG PRINT:
"KLO = ___"
33

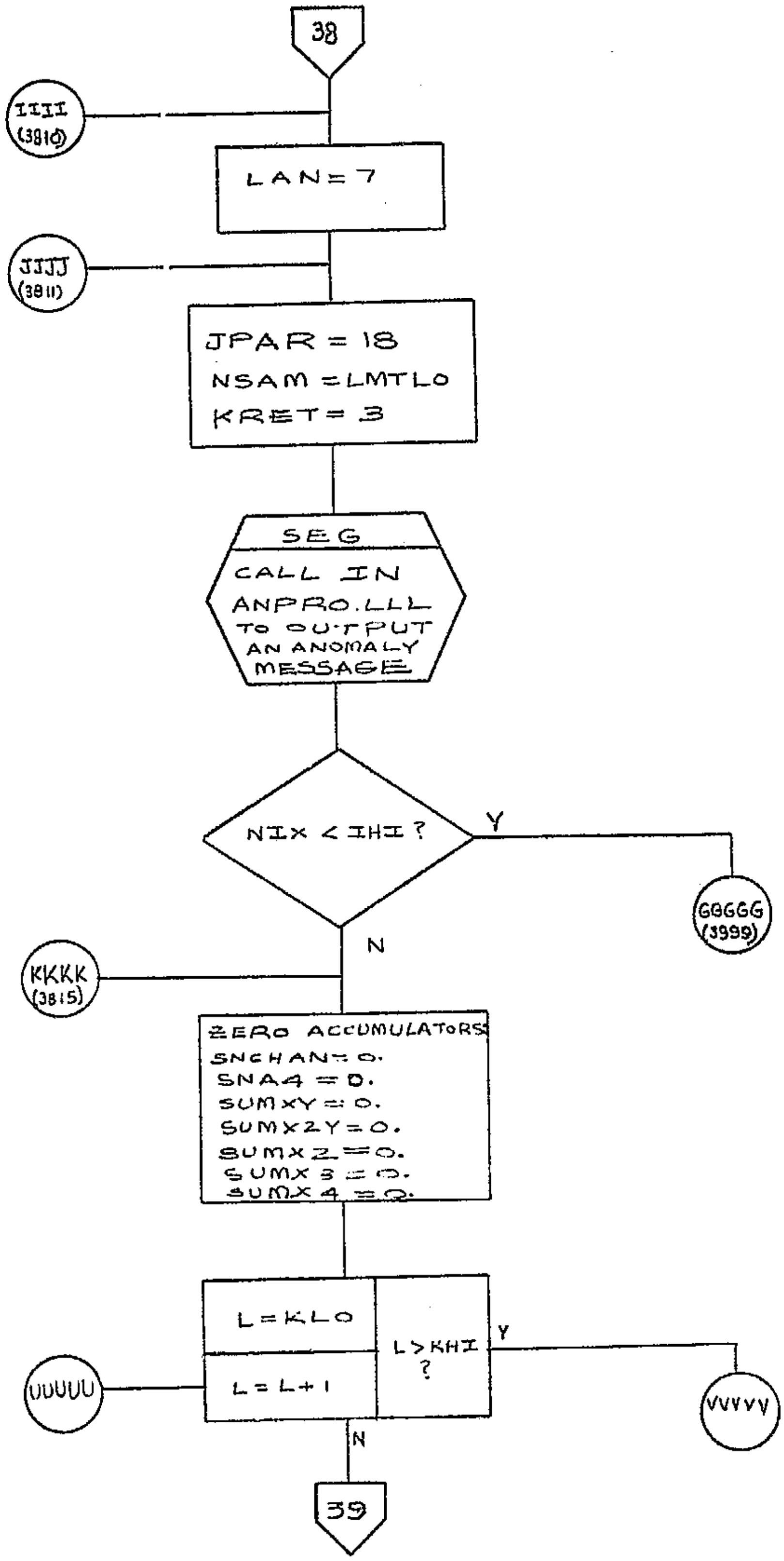

33
ZZZ (3781)
CONTINUE
K = JLO
K = K + 1
K > LMTHI ?
Y
SSSSS
TTTTT
N
NCHAN (K) ≤ INEW ?
Y
BBBB (3790)
N
CONTINUE
SSSSS
TTTTT
INDICATORS
NSAM = LMTLO
LAN = 6
KRET = 3
34

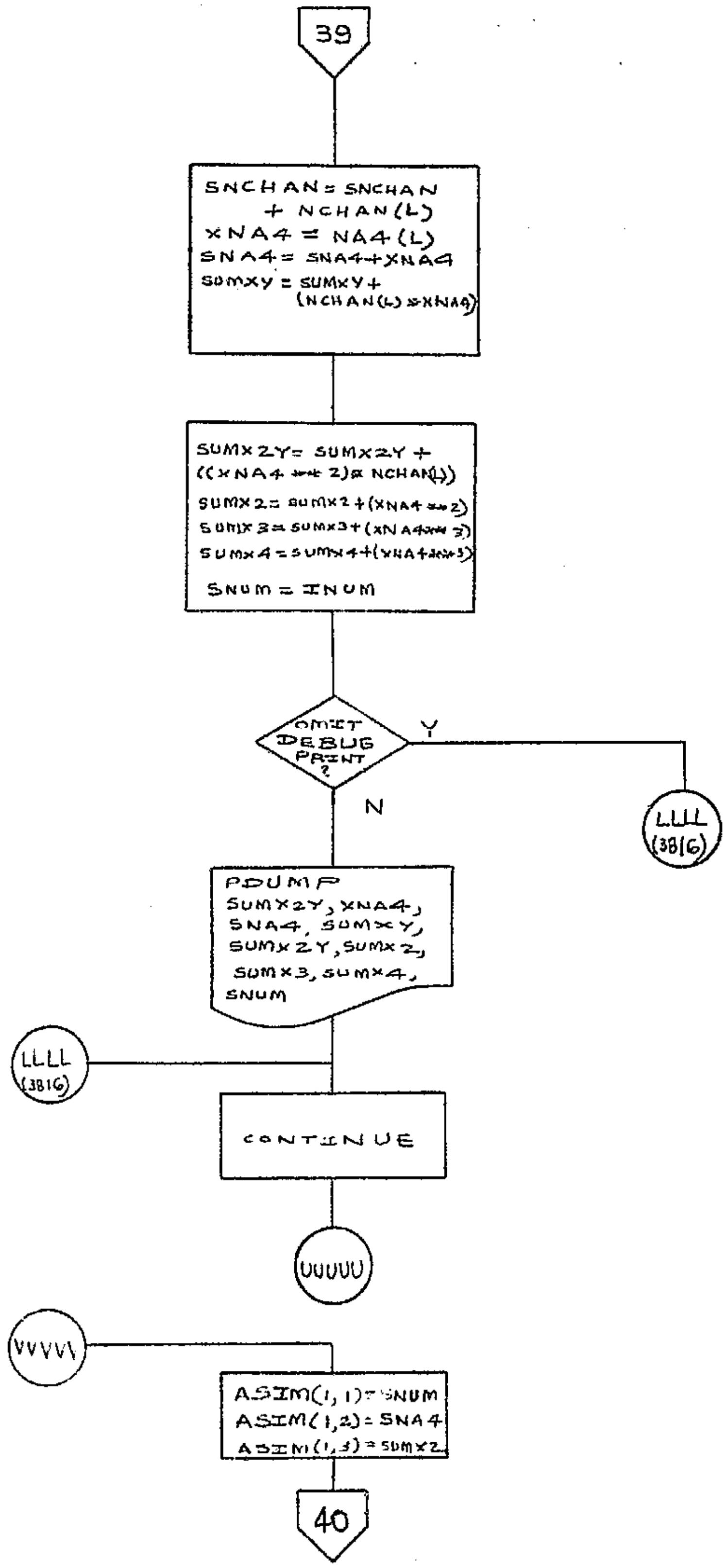

34
SEG
CALL IN
ANPRO.LLL
To OUTPUT
AN ANOMALY
MESSAGE
NIX < IHI
?
Y
H
(2130)
N
GGGGG
(3999)
BBBB
(3790)
KHI = K
OMIT
DEBUG
PRINT
?
Y
DDDD
(3792)
N
DEBUG PRINT:
"KHI = ___"
DDDD
(3792)
CONTINUE
35

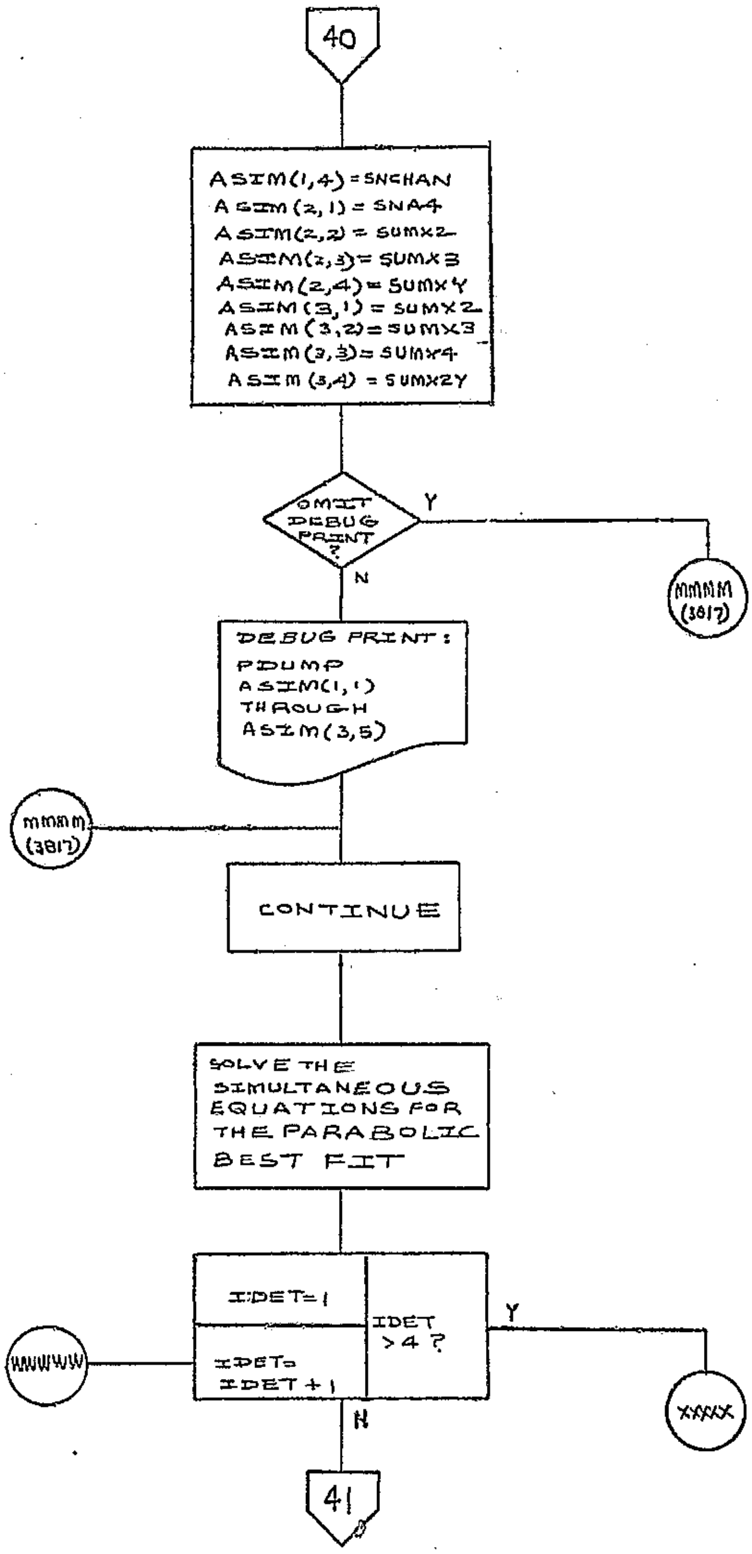

35
KLO > LMTLO ?
Y
EEEE (3795)
N
SET ANPRO.LLL INDICATORS
JPAR=18
NSANI=LMTLO
LAN=4
KRET=3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
NIX < IHI ?
Y
H (2134)
N
GGGG (3999)

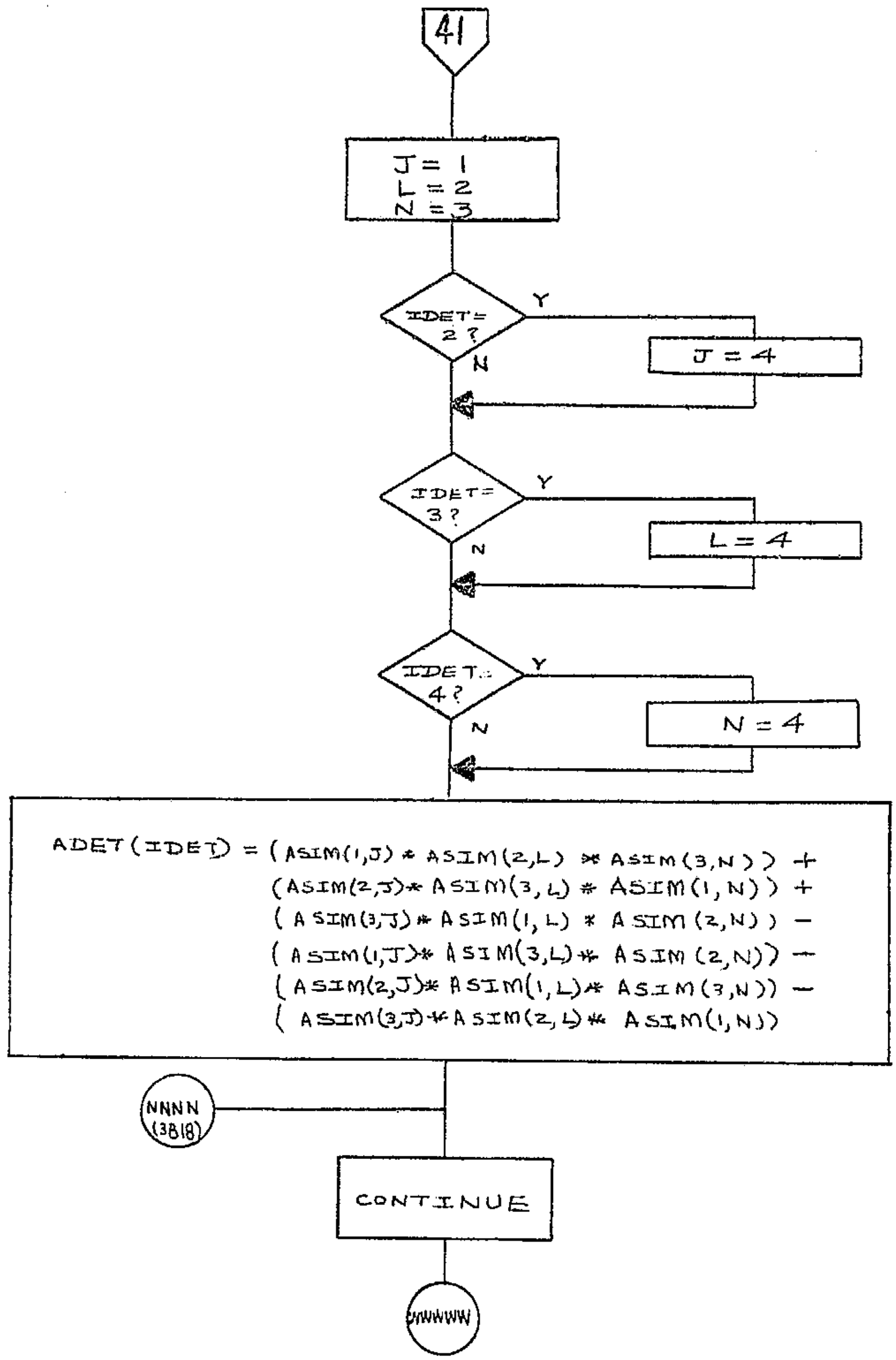

36
EEEE (3795)
KHI < LMTHI
Y
N
FFFF (3800)
SET ANPRO.LLL INDICATORS:
JPAR = 18
NSAM = LMTLO
LAN = 3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
NIX < IHI ?
Y
N
H (2130)
GGGGG (3999)
FFFF (3800)
INUM = KHI - KLO + 1
37

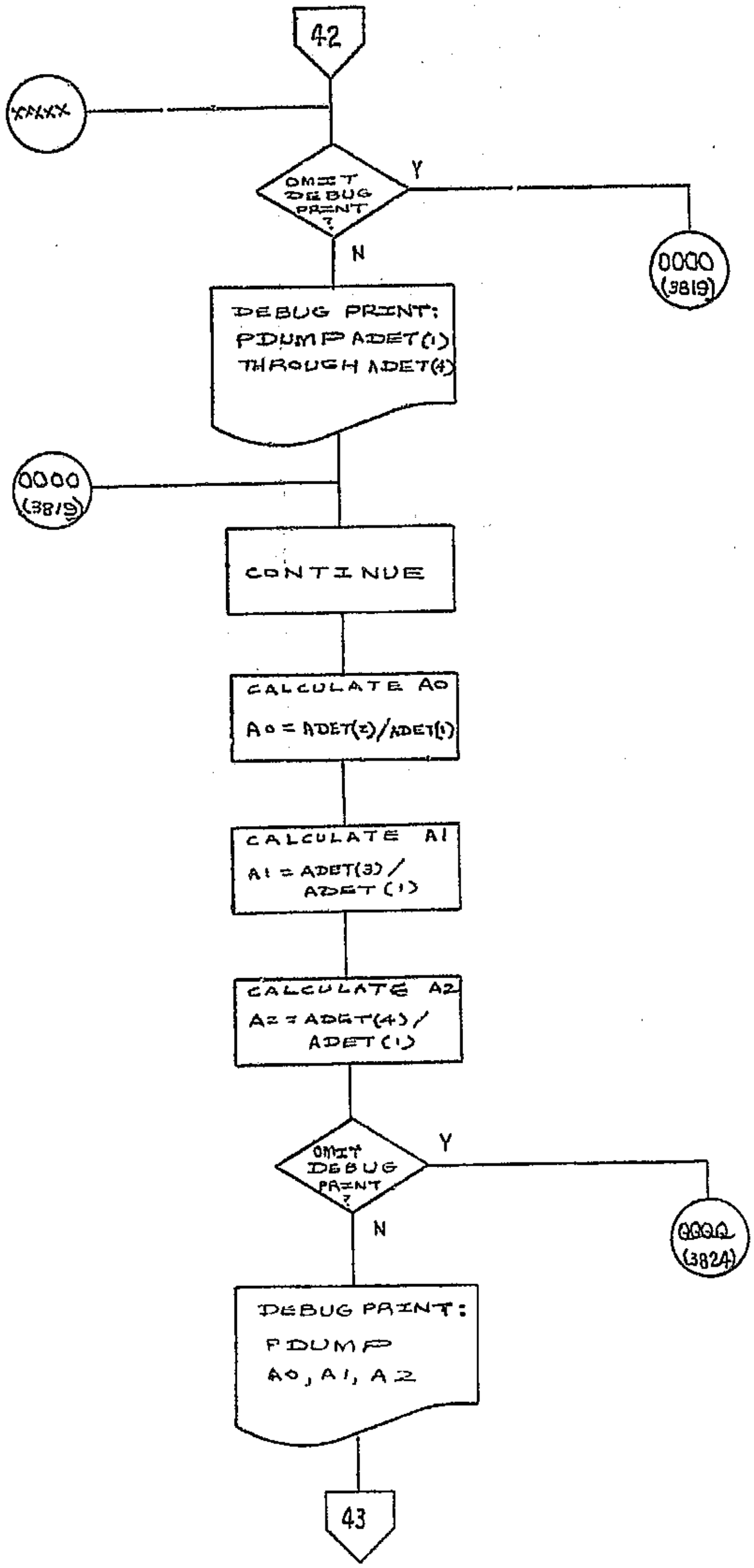

37
OMIT DEBUG PRINT ?
Y
N
GGGG (3801)
DEBUG PRINT:
"INUM = ___"
GGGG (3801)
CONTINUE
INUM < 6 ?
Y
N
HHHH (3805)
INUM > 10 ?
Y
N
IIII (3810)
KKKK (3815)
HHHH (3805)
LAN = 8
JJJJ (3811)

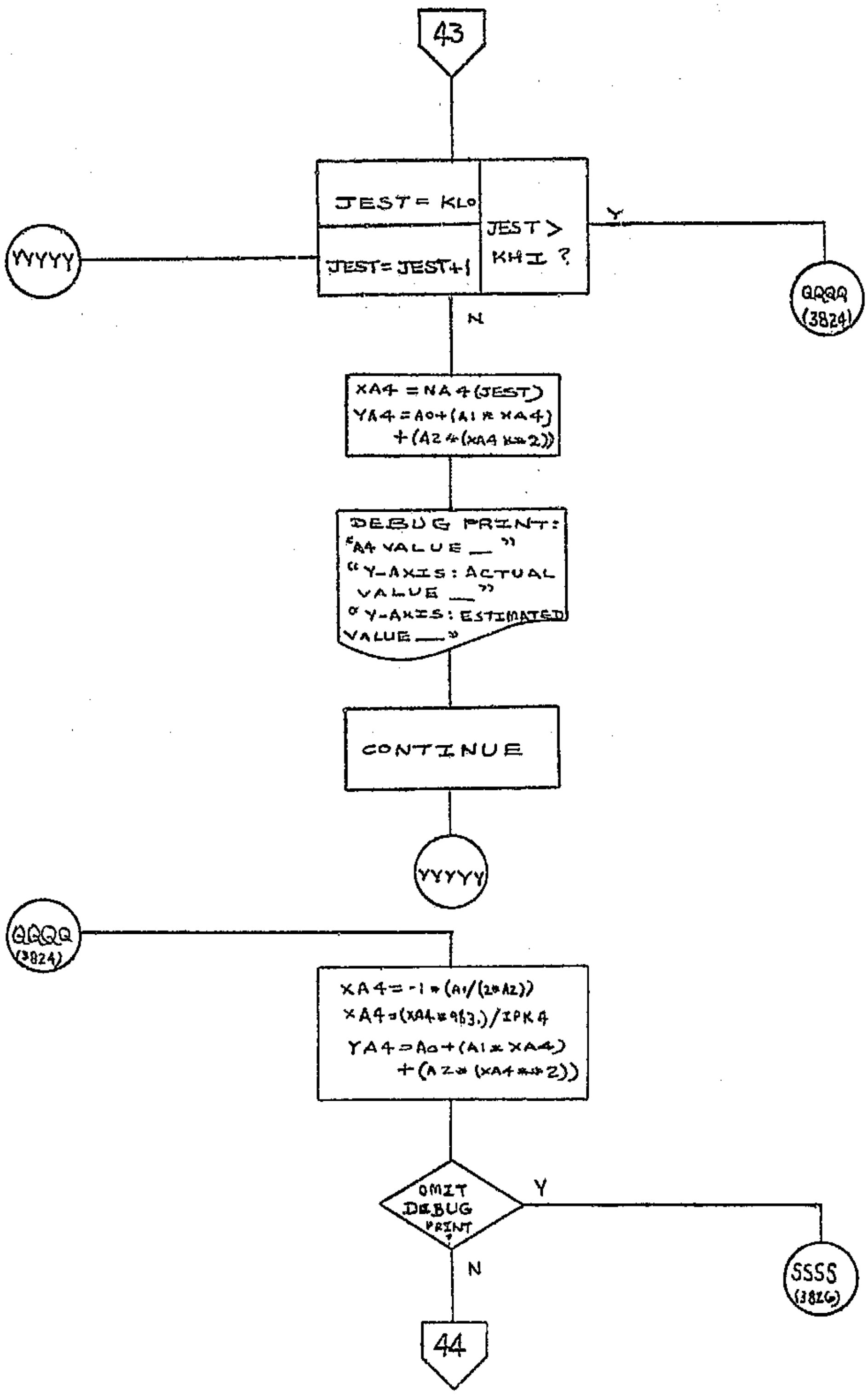

38
IIII (3810)
LAN=7
JJJJ (3811)
JPAR = 18
NSAM = LMTLO
KRET = 3
SEG
CALL IN ANPRO.LLL TO OUTPUT AN ANOMALY MESSAGE
NIX < IHI ?
Y
GGGGG (3999)
N
KKKK (3815)
ZERO ACCUMULATORS
SNCHAN = 0.
SNA4 = 0.
SUMXY = 0.
SUMX2Y = 0.
SUMX2 = 0.
SUMX3 = 0.
SUMX4 = 0.
L=KLO
L=L+1
L>KHI ?
Y
N
UUUUU
VVVVV
39

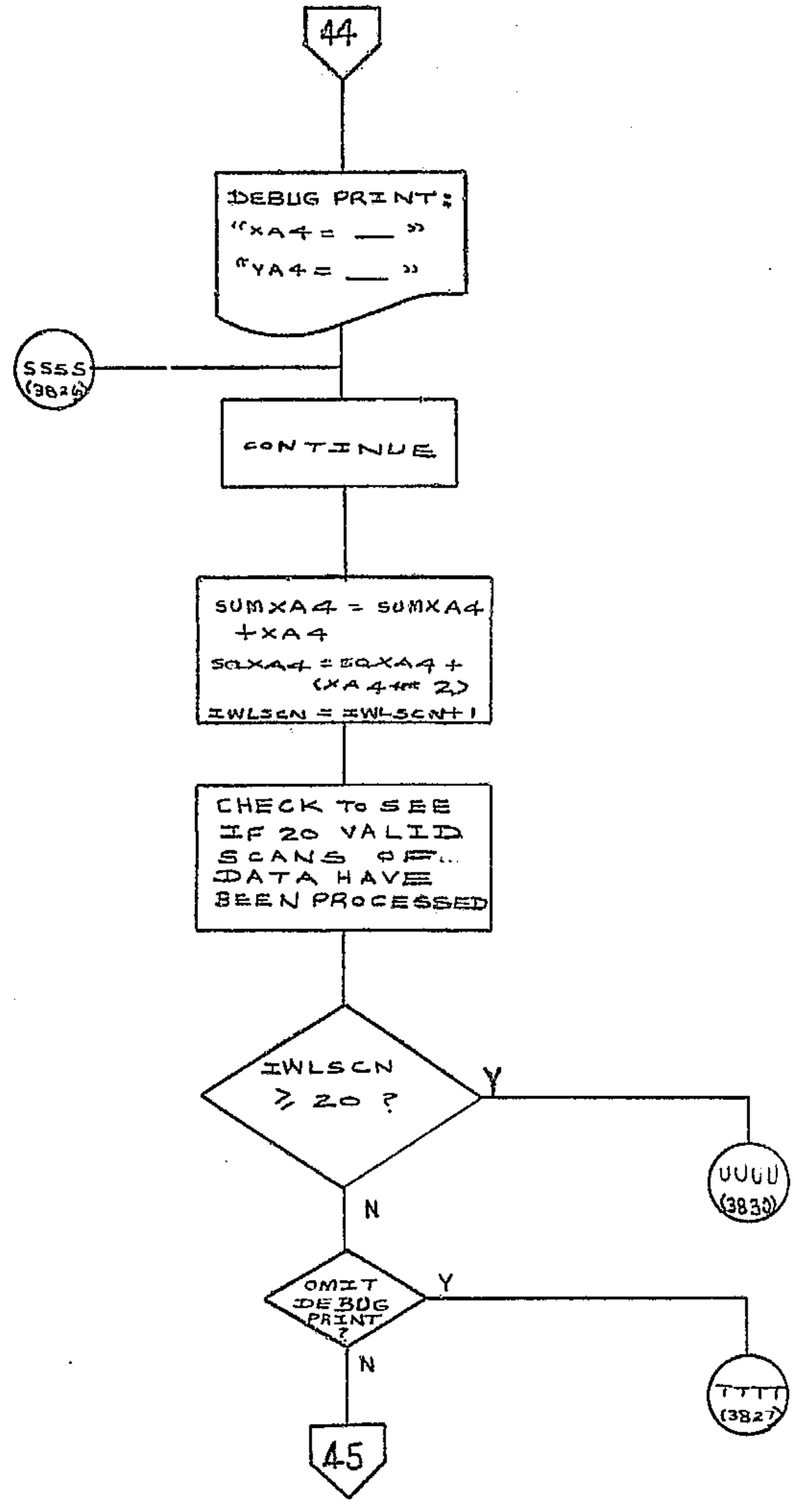

39
SNCHAN = SNCHAN + NCHAN(L)
XNA4 = NA4(L)
SNA4 = SNA4+XNA4
SUMXY = SUMXY + (NCHAN(L)*XNA4)
SUMX2Y = SUMX2Y + ((XNA4**2)*NCHAN(L))
SUMX2 = SUMX2 + (XNA4**2)
SUMX3 = SUMX3 + (XNA4**3)
SUMX4 = SUMX4 + (XNA4**3)
SNUM = INUM
OMIT DEBUG PRINT ?
Y
N
LLLL (3816)
PDUMP SUMX2Y, XNA4, SNA4, SUMXY, SUMX2Y, SUMX2, SUMX3, SUMX4, SNUM
LLLL (3816)
CONTINUE
UUUUU
VVVVV
ASIM(1,1) = SNUM
ASIM(1,2) = SNA4
ASIM(1,3) = SUMX2
40

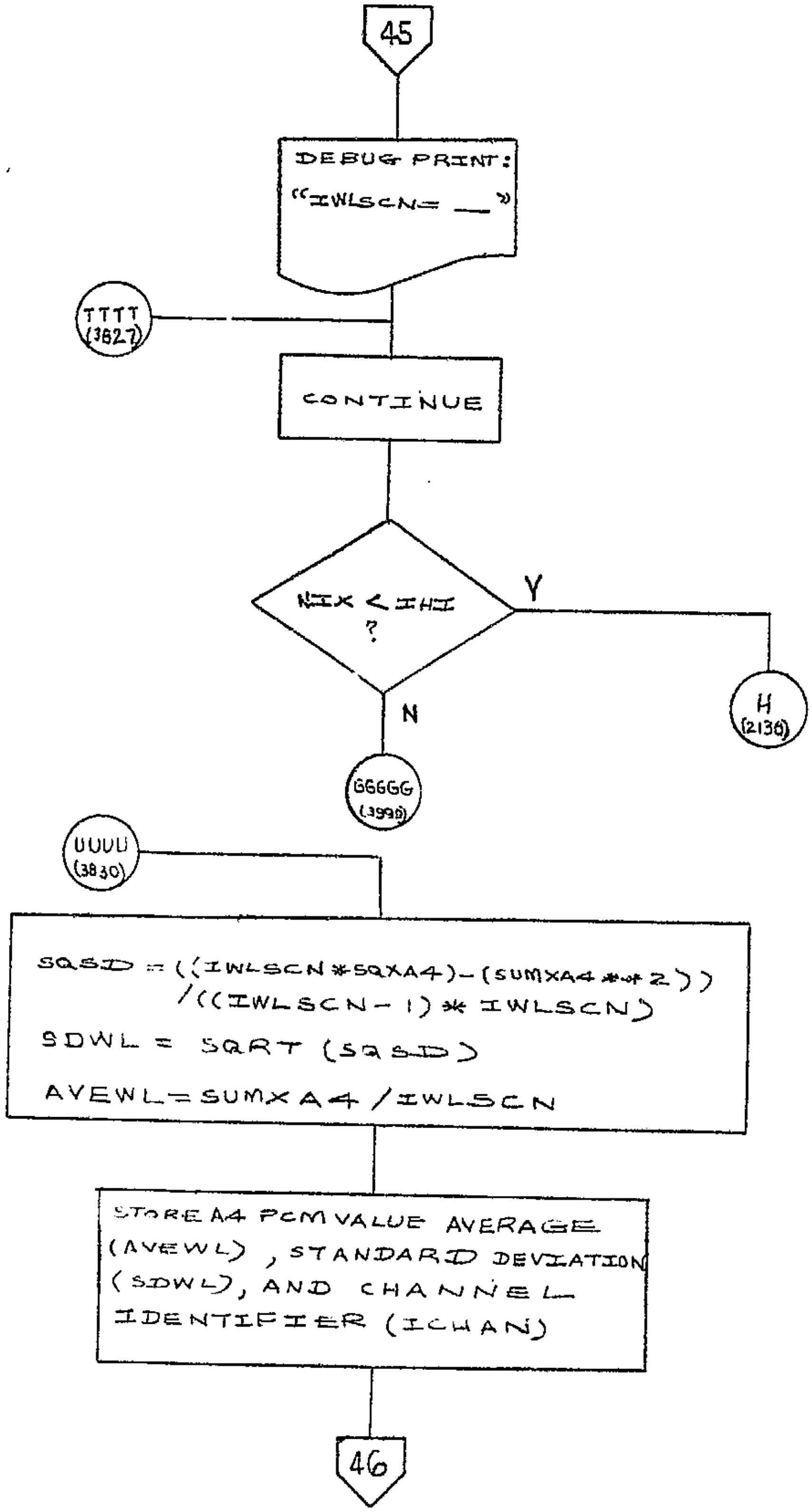

40
ASIM(1,4) = SNCHAN
ASIM(2,1) = SNA4
ASIM(2,2) = SUMX2
ASIM(2,3) = SUMX3
ASIM(2,4) = SUMXY
ASIM(3,1) = SUMX2
ASIM(3,2) = SUMX3
ASIM(3,3) = SUMX4
ASIM(3,4) = SUMX2Y
OMIT DEBUG PRINT ?
Y
N
MMMM (3817)
DEBUG PRINT: PDUMP ASIM(1,1) THROUGH ASIM(3,5)
MMMM (3817)
CONTINUE
SOLVE THE SIMULTANEOUS EQUATIONS FOR THE PARABOLIC BEST FIT
IDET=1
IDET= IDET+1
IDET >4 ?
Y
WWWW
XXXX
N
41

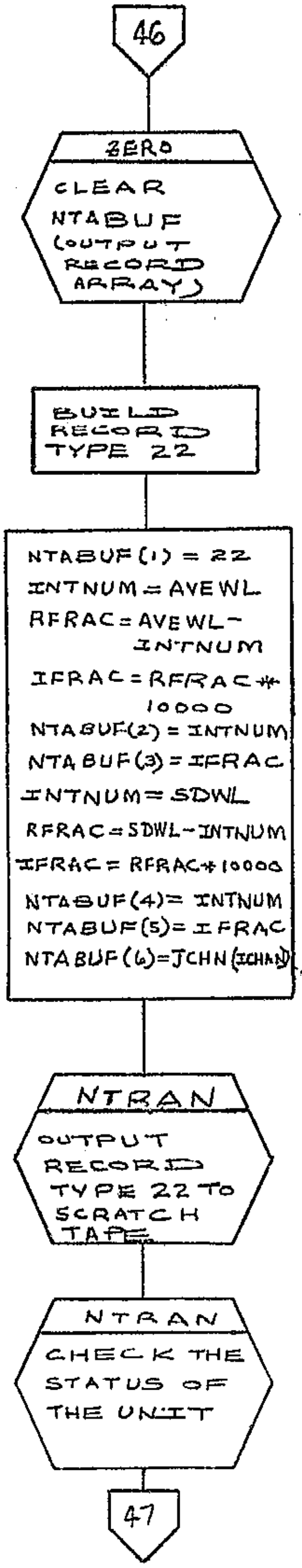

41
J = 1
L = 2
N = 3
IDET = 2 ?
Y
N
J = 4
IDET = 3 ?
Y
N
L = 4
IDET = 4 ?
Y
N
N = 4
ADET(IDET) = (ASIM(1,J) * ASIM(2,L) * ASIM(3,N)) +
(ASIM(2,J) * ASIM(3,L) * ASIM(1,N)) +
(ASIM(3,J) * ASIM(1,L) * ASIM(2,N)) −
(ASIM(1,J) * ASIM(3,L) * ASIM(2,N)) −
(ASIM(2,J) * ASIM(1,L) * ASIM(3,N)) −
(ASIM(3,J) * ASIM(2,L) * ASIM(1,N))
NNNN
(3818)
CONTINUE
WWWWW

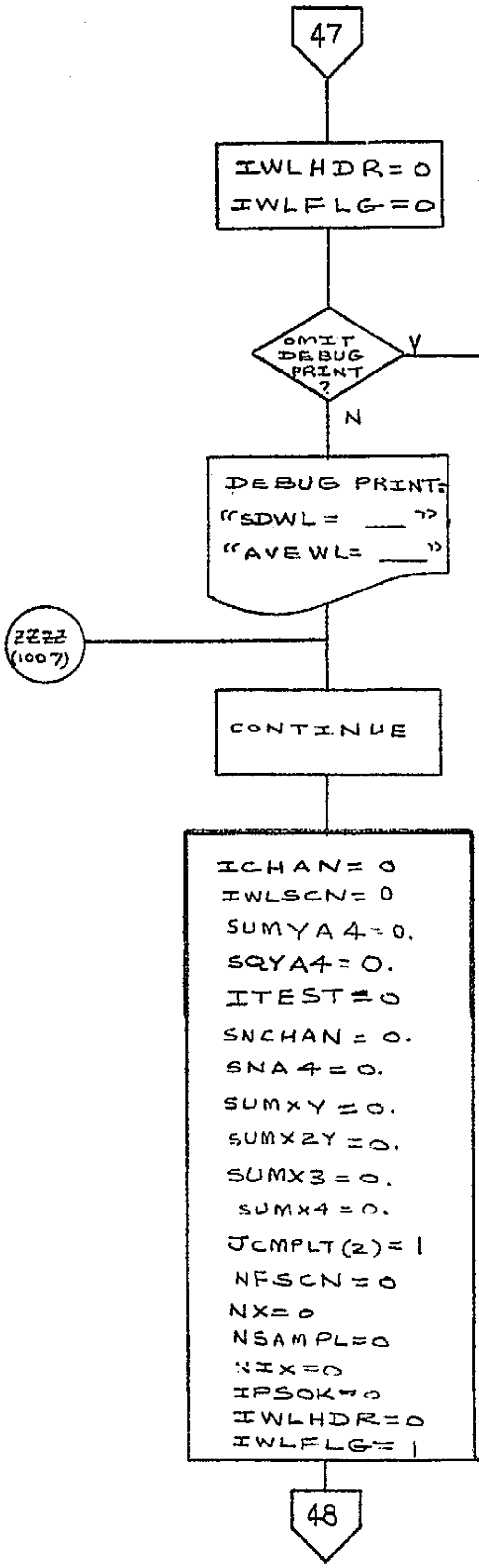

42
XXXX
OMIT DEBUG PRINT ?
Y
N
0000 (3819)
DEBUG PRINT: PDUMP ADET(1) THROUGH ADET(4)
0000 (3819)
CONTINUE
CALCULATE A0
A0 = ADET(2)/ADET(1)
CALCULATE A1
A1 = ADET(3) / ADET (1)
CALCULATE A2
A2 = ADET(4) / ADET (1)
OMIT DEBUG PRINT ?
Y
N
QQQQ (3824)
DEBUG PRINT: PDUMP A0, A1, A2
43

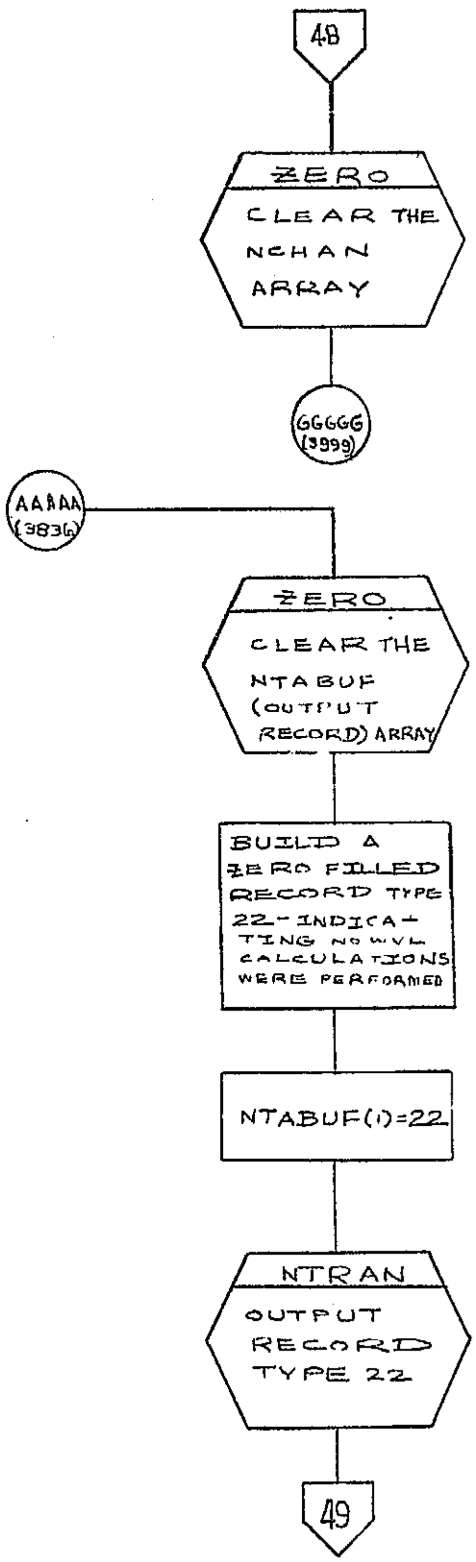

43
JEST = KLO
JEST = JEST+1
JEST > KHI ?
Y
N
YYYY
QQQQ (3824)
XA4 = NA4(JEST)
YA4 = A0+(A1 * XA4)
+(A2*(XA4**2))
DEBUG PRINT:
"A4 VALUE __"
"Y-AXIS: ACTUAL VALUE __"
"Y-AXIS: ESTIMATED VALUE __"
CONTINUE
YYYYY
QQQQ (3824)
XA4 = -1 * (A1/(2*A2))
XA4 = (XA4*963.)/IPK4
YA4 = A0+(A1 * XA4)
+(A2* (XA4**2))
OMIT DEBUG PRINT ?
Y
N
SSSS (3826)
44

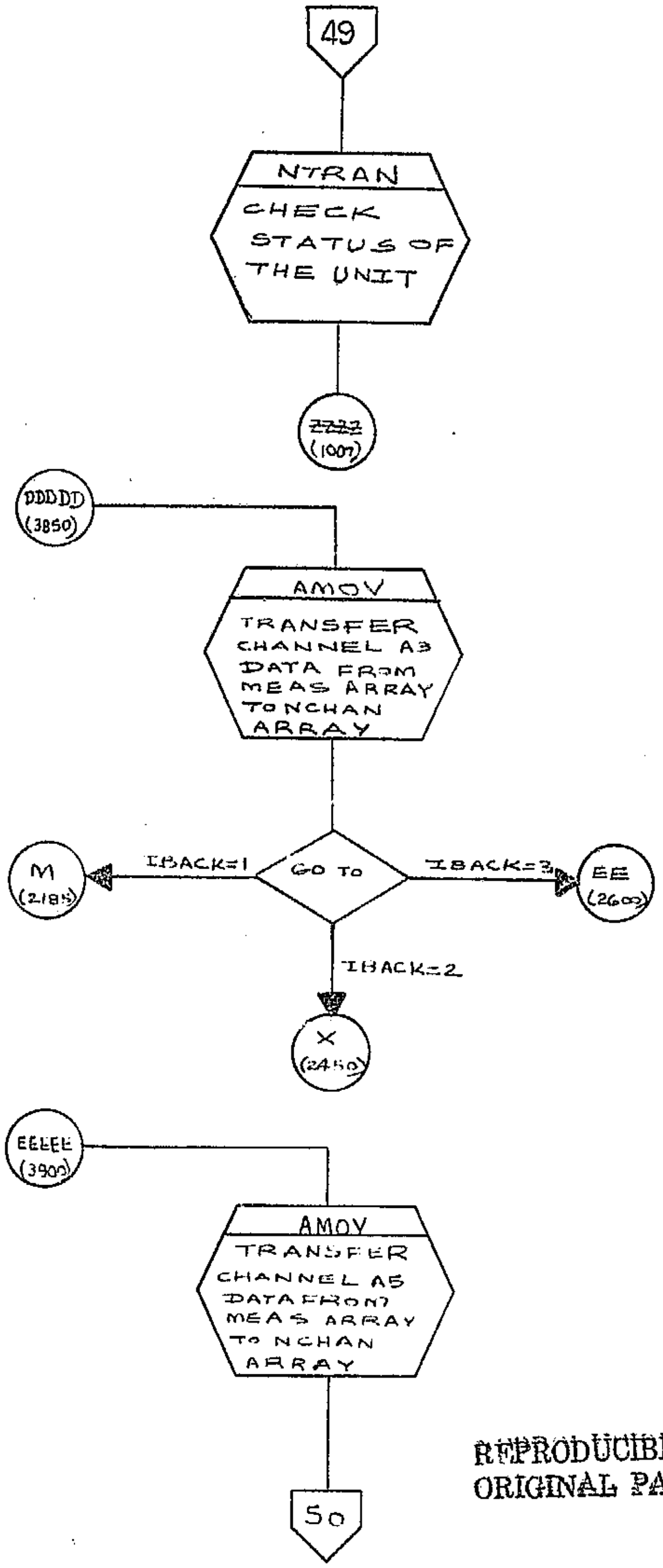

44
DEBUG PRINT:
"XA4= ___"
"YA4= ___"
SSSS
(3826)
CONTINUE
SUMXA4 = SUMXA4 +XA4
SQXA4 = SQXA4 + (XA4** 2)
IWLSCN = IWLSCN+1
CHECK TO SEE IF 20 VALID SCANS OF DATA HAVE BEEN PROCESSED
IWLSCN ≥ 20 ?
Y
UUUU
(3830)
N
OMIT DEBUG PRINT ?
Y
TTTT
(3827)
N
45

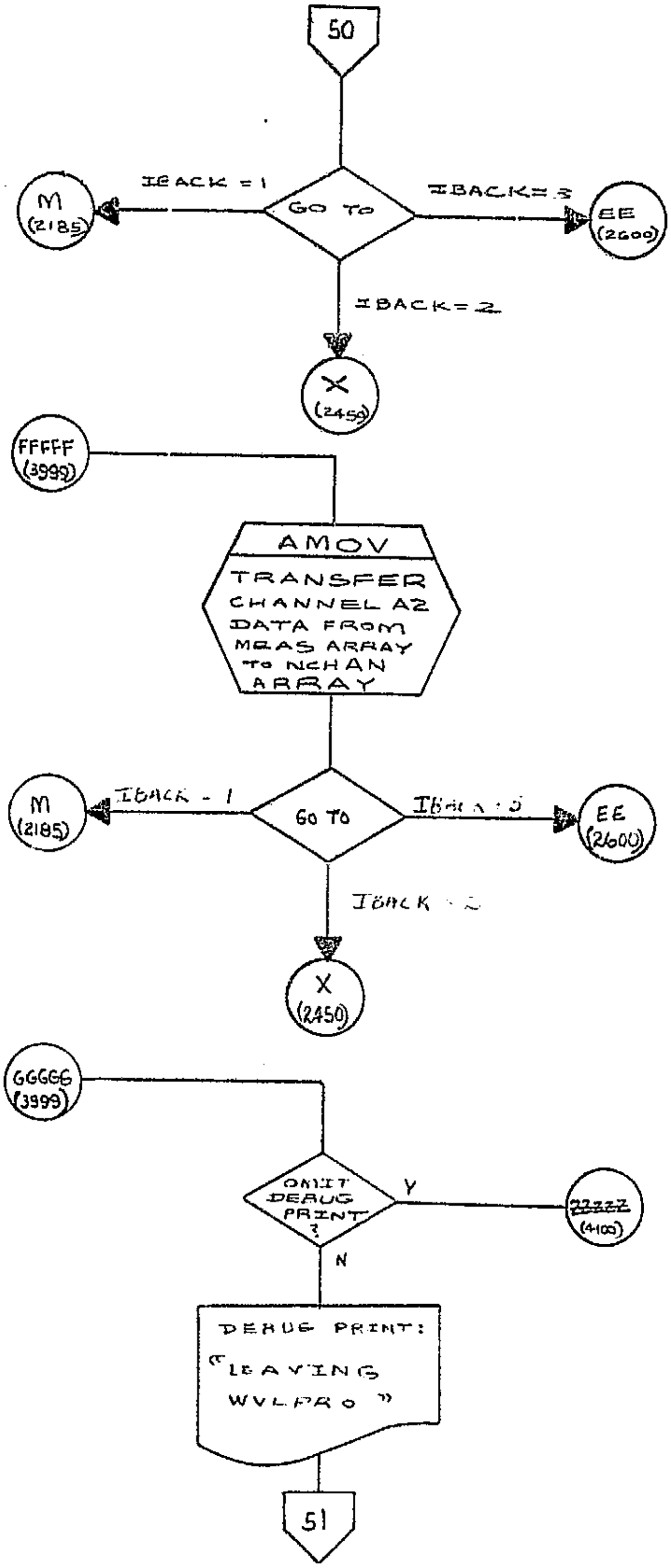

45
DEBUG PRINT:
"IWLSCN= __"
TTTT
(3827)
CONTINUE
NIX < IHI
?
Y
N
H
(2136)
GGGGG
(3998)
UUUU
(3830)
SQSD = ((IWLSCN*SQXA4)-(SUMXA4**2))
/((IWLSCN-1)*IWLSCN)
SDWL = SQRT (SQSD)
AVEWL=SUMXA4/IWLSCN
STORE A4 PCM VALUE AVERAGE
(AVEWL), STANDARD DEVIATION
(SDWL), AND CHANNEL
IDENTIFIER (ICHAN)
46

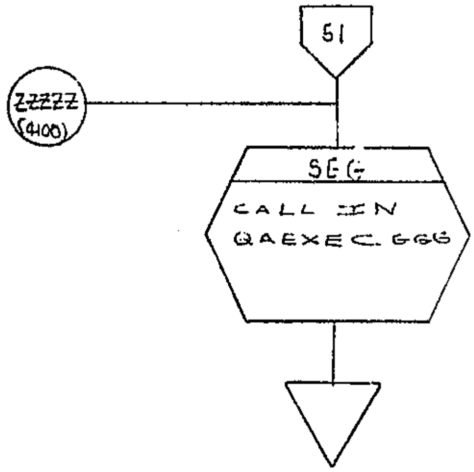

46
ZERO
CLEAR NTABUF (OUTPUT RECORD ARRAY)
BUILD RECORD TYPE 22
NTABUF(1) = 22
INTNUM = AVEWL
RFRAC = AVEWL-INTNUM
IFRAC = RFRAC * 10000
NTABUF(2) = INTNUM
NTABUF(3) = IFRAC
INTNUM = SDWL
RFRAC = SDWL-INTNUM
IFRAC = RFRAC * 10000
NTABUF(4) = INTNUM
NTABUF(5) = IFRAC
NTABUF(6) = JCHN(ICHAN)
NTRAN
OUTPUT RECORD TYPE 22 TO SCRATCH TAPE
NTRAN
CHECK THE STATUS OF THE UNIT
47


47
IWLHDR=0
IWLFLG=0
OMIT DEBUG PRINT ?
Y
N
ZZZZ (1007)
DEBUG PRINT:
"SDWL = ___"
"AVEWL= ___"
ZZZZ (1007)
CONTINUE
ICHAN= 0
IWLSCN= 0
SUMYA4=0.
SQYA4= 0.
ITEST=0
SNCHAN = 0.
SNA4=0.
SUMXY =0.
SUMX2Y =0.
SUMX3 =0.
SUMX4 =0.
JCMPLT(2)= 1
NFSCN=0
NX=0
NSAMPL=0
NIX=0
IPSOK=0
IWLHDR=0
IWLFLG= 1
48


48
ZERO
CLEAR THE NCHAN ARRAY
GGGGG (3999)
AAAAA (3836)
ZERO
CLEAR THE NTABUF (OUTPUT RECORD) ARRAY
BUILD A ZERO FILLED RECORD TYPE 22-INDICATING NO WVL CALCULATIONS WERE PERFORMED
NTABUF(1)=22
NTRAN
OUTPUT RECORD TYPE 22
49


49
NTRAN
CHECK STATUS OF THE UNIT
(1007)
DDDDD
(3850)
AMOV
TRANSFER CHANNEL A3 DATA FROM MEAS ARRAY TO NCHAN ARRAY
GO TO
IBACK=1
M
(2185)
IBACK=3
EE
(2600)
IBACK=2
X
(2450)
EEEEE
(3900)
AMOV
TRANSFER CHANNEL A5 DATA FROM MEAS ARRAY TO NCHAN ARRAY
50


50
GO TO
IBACK = 1
IBACK = 3
IBACK = 2
M
(2185)
EE
(2600)
X
(2450)
FFFFF
(3999)
AMOV
TRANSFER CHANNEL A2 DATA FROM MEAS ARRAY TO NCHAN ARRAY
GO TO
IBACK = 1
IBACK = 3
IBACK = 2
M
(2185)
EE
(2600)
X
(2450)
GGGGG
(3999)
OMIT DEBUG PRINT ?
Y
N
ZZZZZ
(4100)
DEBUG PRINT:
"LEAVING WVLPRO"
51


51
ZZZZZ
(4100)
SEG
CALL IN
QAEXEC.GGG

2.6 STORAGE REQUIREMENTS

The storage requirements for QA191H may be determined by referring to the program listing on 2.10.

2.7 NAMED COMMONS

Figure 2.7-1 is a cross reference of the named commons used by the QA191H program with the resident program, load modules, data areas, and subroutines.

Named Common Name	Length Decimal Words	Length Octal Bytes	Resident	QA191H	Data Areas	DCMBUF	Load Modules	ANPRO	BIAPRO	CONDRV	DCMDRY	DECRIP	ERRDRV	FLDPRO	QAEXEC	QASUM	RAMPRO	RESPRO	TMLOOP	WVLPRO
A4CHNL	685	2532		X													X			X
ANDAT	91	266		X				X	X					X	X	X	X	X		X
BIADAT	38	114		X					X									X		
CALRT	3	6		X										X	X	X				
CMPLET	2	4		X											X	X	X			
DCARGN	28	70		X				X			X	X		X	X	X	X	X	X	
DCDATA	3993	17462		X		X		X	X		X	X		X	X	X	X	X	X	X
DCCNTL	88	260										X								
ERROR	20	50		X						X	X	X	X		X	X				
ERROR1	11	26											X							
FLDAT	4	10		X										X						
HISDAT	8	20		X					X						X	X	X	X		X
INPUT	158	474		X				X	X	X			X	X	X	X	X	X	X	X
INTNDX	5	12		X				X	X						X	X	X	X	X	X
LASFRM	4	10		X											X					
PRNDX	54	154		X					X						X	X	X	X	X	X
QADAT	5	12		X										X	X	X				
RDARG	6	14								X										
RDCNTL	780	3030								X										
R[illegible]PTR	1	2		X											X			X		
RESDAT	712	2620		X					X									X		X
SAVE	33	102		X				X									X			
SIXSV	1	2		X					X									X		X
TIMES	522	2024		X				X	X	X	X	X			X	X	X	X	X	X
TITLES	32	100		X											X	X	X			
WVLDAT	4	10		X																X

Figure 2.7-1 Cross Reference of Named Commons with the Resident program, Load Modules, Data Areas, and Subroutines for QA191H.

Table 2.7-1 Layout of Named Common A4CHNL

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-685	IA4	685	SPI	PCM COUNTS	STORAGE AREA FOR A SCAN LINE OF CHANNEL A4 DATA

Table 2.7-2 Layout of Named Common ANDAT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	JPAR	1	SPI		PARAMETER CODE FOR ANOMALY PROCESSOR (ANPRO)
2	NSAM	1	SPI		POINTER FOR ANOMALOUS SAMPLE IN MEAS ARRAY
3	LAN	1	SPI		FAILURE CODE FOR ANOMALY PROCESSOR (ANPRO)
4	MACT	1	SPI		ACTIVE ANOMALY COUNTER FLAG
5-19	ANSTAT	15	SPI		STATUS INDICATOR FOR ANOMALY COUNTERS
20	KRET	1	SPI		FLAG FOR RETURNING PROGRAM CONTROL FROM ANPRO
21	ANCNTR	15	SPI		ANOMALY COUNTERS FOR ANOMALY SUB-TOTALS
36-50	ANREC	15	SPI		RECORD POINTER TO FLAG ANOMALY SUB-TOTAL PRINT
51-86	TANCTR	18	SPR		ANOMALY COUNTERS FOR ANOMALY GRAND TOTALS
87	IRESFG	1	SPI		FLAG FOR INDICATING ABORTED RESPONSIVITY CALCULATIONS
88	IWLFG	1	SPI		FLAG FOR INDICATING ABORTED W/L CAL CALCULATIONS
89	IRAMFG	1	SPI		FLAG FOR INDICATING ABORTED W/L RAMP CALCULATIONS
90	LZNE	1	SPI		LINE COUNTER FOR ANOMALY PRINT OUT
91	IPG	1	SPI		PAGE COUNTER FOR ANOMALY PRINT OUT

Table 2.7-3 Layout of Named Common BIADAT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	IBHIS1	7	SPI		ACCUMULATORS FOR CH. A1 HISTOGRAM COUNTS
8 -14	IBHIS2	7	SPI		ACCUMULATORS FOR CH. A2 HISTOGRAM COUNTS
15 -21	IBHIS3	7	SPI		ACCUMULATORS FOR CH. A3 HISTOGRAM COUNTS
22 -28	IBHIS5	7	SPI		ACCUMULATORS FOR CH. A5 HISTOGRAM COUNTS
29 -35	IBHIS6	7	SPI		ACCUMULATORS FOR CH. A6 HISTOGRAM COUNTS
36	ISSN	1	SPI		SCAN COUNTER FOR BIAS VOLTAGE HISTOGRAMS
37	LCMPLT	2	SPI		COMPLETION STATUS FLAG FOR BIAPRO

Table 2.7-4 Layout of Named Common CAIRT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	JJNT	1	SPI		FLAG FOR INDICATING HEATED CAL RESPONSIVITY COMPLETED
2	NDICAL	1	SPI	PCM	DICHROIC TEMPERATURE EXTRACTED FROM HEATED CAL
3	NTRCAL	1	SPI	PCM	THERMAL REF. TEMPERATURE EXTRACTED FROM HEATED CAL

Table 2.7-5 Layout of Named Common CMPLET

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	NCMPLT(1)	1	SPI		NUMBER INDICATING LAST TIME PERIOD PROCESSED
2	NCMPLT(2)	1	SPI		0-PROCESSING HAS NOT BEEN COMPLETED FOR THE TIME PERIOD
					INDICATED BY NCMPLT(1)
					1-PROCESSING HAS BEEN COMPLETED FOR THE TIME PERIOD
					INDICATED BY NCMPLT(1)

Table 2.7-6 Layout of Named Common DCARGN

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-28	DCARGN				THE NAMED COMMON DCARGN, WHICH FORMS THE INTERFACE FOR THE TAPE INPUT ROUTINES DECRIP AND DCMDRV, IS DESCRIBED IN SECTION 2.2.4.2 OF DOCUMENT NUMBER ERS-300-02, "EARTH RESOURCES DATA DECOMMUTATIO PROGRAMS".

T 2.7-7

LAYOUT OF NAMED COMMON DCDATA FOR QA191H

	SAMPLE RATE (SAMPLES//FRAME)	MEASUREMENT NUMBER	NO. OF SAMPLES PER MEAS ARRAY	MEAS ARRAY INDICES: TIME	DELTA TIME	WORD INDEX	NO. OF VALUES	NO. OF 16-BIT LOCATIONS REQ.	FIRST DATA VALUE LOCATION
1.	1	D007-RRO	18	1	85	169	190	211	232
2.	1	A016-RRO	18	5	89	170	191	212	250
3.	1	A018-RRO	18	9	93	171	192	213	268
4.	1	A007-RRO	18	13	97	172	193	214	286
5.	1	A008-RRO	18	17	101	173	194	215	304
6.	1	A019-RRO	18	21	105	174	195	216	322
7.	1	A015-RRO	18	25	109	175	196	217	340
8.	1	A009-RRO	18	29	113	176	197	218	358
9.	1	A013-RRO	18	33	117	177	198	219	376
10.	1	A014-RRO	18	37	121	178	199	220	394
11.	1	D005-RRO	18	41	125	179	200	221	412
12.	1	D006-RRO	18	45	129	180	201	222	430
13.	2	A020-RRO	36	49	133	181	202	223	448
14.	2	A017-RRO	36	53	137	182	203	224	484
15.	32	A001-RRO	576	57	141	183	204	225	520
16.	32	A002-RRO	576	61	145	184	205	226	1096

TABLE 2.7-7

LAYOUT OF NAMED COMMON DCDATA FOR QA191H

	SAMPLE RATE (SAMPLES/FRAME)	MEASUREMENT NUMBER	NO. OF SAMPLES PER MEAS ARRAY	MEAS ARRAY INDICES: TIME	DELTA TIME	WORD INDEX	NO. OF VALUES	NO. OF 16-BIT LOCATIONS REQ.	FIRST DATA VALUE LOCATION
17.	32	A003-RRO	576	65	149	185	206	227	1672
18.	32	A004-RRO	576	69	153	186	207	228	2248
19.	32	A005-RRO	576	73	157	187	208	229	2824
20.	32	A006-RRO	576	77	161	188	209	230	3400
21.	1	A023-RRO	18	81	165	189	210	231	3976

(FOR ADDITIONAL INFORMATION REFER TO EARTH RESOURCES DATA DECOMMUTATION PROGRAMS (DECOM1, DCRIPT, DCOM2N, DCOM2I) PROGRAM DOCUMENT, ERS-300-02)

Table 2.7-8 Layout of Named Common DCCNTL

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	ISENSR	1	SPI		THE NAMED COMMON DCCNTL PROVIDES THE INTERFACE DATA FOR TAPE ROUTINE
2	NTPS	1	SPI		DCMDRV AND IS DESCRIBED IN SECTION 2.2.4.2 IN DOCUMENT NUMBER
3	IDDEC	1	SPI		ERS-300-02, "EARTH RESOURCES DATA DECOMMUTATION PROGRAMS".
4	NM	1	SPI		
5-88	MIDH	84	FA		

Table 2.7-9 Layout of Named Common ERROR

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-20	IERR	20	SPI		ERROR IS UTILIZED IN INTERFACING WITH THE ERROR PROCESSOR ERRPRC, AND IS DEFINED IN SECTION 2.5.3-B OF DOCUMENT NUMBER ERS-300-07, "EARTH RESOURCES APPLICATION UTILITY ROUTINES".

Table 2.7-10 Layout of Named Common ERROR1

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	N	1	SPI		
2-11	IUARRY	10	SPI		ERROR1 IS UTILIZED IN INTERFACING WITH ERROR PROCESSOR ERRDRV AND IS DEFINED IN SECTION 2.5.3 B OF DOCUMENT NUMBER ERS-300-07, "EARTH RESOURCES APPLICATION UTILITY ROUTINES".

Table 2.7-11 Layout of Named Common FLDAT

for

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	JZ	1	SPI		INDEXER FOR MEAS ARRAY DURING PARAMETER CHECKS
2	KZ	1	SPI		SAMPLE COUNTER FOR MEAS ARRAY FIELD DATA
3	LZF	1	SPI		MEAS ARRAY INDEX POINTER
4	NZM	1	SPI		MEAS ARRAY INDEX POINTER

Table 2.7-12 Layout of Named Common HISDAT .

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	IRMFLG	1	SPI		COMPLETION FLAG FOR WRITING W/L RAMP HISTORICAL FILE
2	IRSFLG	1	SPI		COMPLETION FLAG FOR WRITING RESPONSIVITY HISTORICAL FILE
3	IBVFLG	1	SPI		COMPLETION FLAG FOR WRITING BIAS VOLTAGE HISTORICAL FILE
4	IWLFLG	1	SPI		COMPLETION FLAG FOR WRITING W/L CAL HISTORICAL FILE
5	ICLHDR	1	SPI		COMPLETION FLAG FOR WRITING CAL PERIOD HEADER
6	IWLHDR	1	SPI		COMPLETION FLAG FOR WRITING W/L CAL PERIOD HEADER
7	IDONE	1	SPI		INTERIM COUNTER FOR WRITING RESPONSIVITY DATA
8	IBVHDR	1	SPI		COMPLETION FLAG FOR WRITING BIAS VOLTAGE PERIOD HEADER

Table 2.7-13 Layout of Named Common INPUT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-102	IIN	102	SPI		ARRAY USED FOR STORAGE OF INTEGER CONTROL CARD PARAMETERS
103-158	RIN	28	SPI		ARRAY USED FOR STORAGE OF FLOATING POINT CONTROL CARD PARAMETERS
1	IIN(1)	1	SPI		SENSOR CODE (=15)
2	IIN(2)	1	SPI		RECORDING FORM (=27)
3	IIN(3)	1	SPI		MISSION NUMBER (NEGATIVE FOR DEBUG)
4	IIN(4)	1	SPI		FLIGHT NUMBER
5	IIN(5)	1	SPI		SITE NUMBER
6	IIN(6)	1	SPI		LINE NUMBER
7	IIN(7)	1	SPI		RUN NUMBER
8	IIN(8)	1	SPI		BASE YEAR OF DATA RECORDING DATE (LAST 2 DIGITS)
9	IIN(9)	1	SPI		BASE MONTH OF DATA RECORDING DATE
10	IIN(10)	1	SPI		BASE DAY OF DATA RECORDING DATE
11	IIN(11)	1	SPI		0-TAB OUT NEW HISTORICAL FILE
					1-TAB OUT OLD AND NEW HISTORICAL FILE
					2-TAB OLD HISTORICAL FILE, ONLY (NO PROCESSING OF FIELD OR
					CALIBRATION DATA)
12	IIN(12)	1	SPI		0-FILE DELETION IS NOT REQUIRED
					1-FILE DELETION IS REQUIRED
13	IIN(13)	1	SPI	HOURS	TIME OF FILE TO BE DELETED (HOURS SEGMENT)
14	IIN(14)	1	SPI	MINUTES	TIME OF FILE TO BE DELETED (MINUTES SEGMENT)
15	IIN(15)	1	SPI	SECONDS	TIME OF FILE TO BE DELETED (SECONDS SEGMENT)

Table 2.7-13 Layout of Named Common INPUT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
16	IIN(16)	1	SPI		NUMBER OF OVERALL TIME PERIODS TO PROCESS
17	IIN(17)	1	SPI		NUMBER OF CAL PERIODS TO PROCESS
18	IIN(18)	1	SPI		NUMBER OF WAVELENGTH CAL PERIODS TO PROCESS
19	IIN(19)	1	SPI		TOTAL NUMBER OF PERIODS TO PROCESS
20	IIN(20)	1	SPI		NUMBER OF CONSECUTIVE CAL PERIOD SCANS TO PROCESS FOR WAVELENGTH RAMP CALCULATIONS
21	IIN(21)	1	SPI	PCM COUNTS	THE PLUS OR MINUS TOLERANCE LIMIT FOR PCM COUNT DEVIATION DURING LINEARITY OF SCAN CHECK
22	IIN(22)	1	SPI	PCM COUNTS	MINIMUM PCM COUNT VALUE OF PARAMETERS A1, A2, A3, A5 AND A6 FOR VALID SYNC PULSES
23	IIN(23)	1	SPI	PCM COUNTS	MINIMUM PCM COUNT VALUE OF A4 FOR VALID END OF SCAN
24	IIN(24)	1	SPI		MINIMUM NUMBER OF DATA SAMPLES FOR VALID SCAN
25	IIN(25)	1	SPI		MAXIMUM NUMBER OF DATA SAMPLES FOR VALID SCAN
26-32	IIN(26-IIN(32)	7	SPI	PCM COUNTS	ASSIGNED VALUE RANGES FOR THE A1 BIAS VOLTAGE HISTOGRAM
33-39	IIN(33)-IIN(39)	7	SPI	PCM COUNTS	ASSIGNED VALUE RANGES FOR THE A2 BIAS VOLTAGE HISTOGRAM
40-46	IIN(40)-IIN(46)	7	SPI	PCM COUNTS	ASSIGNED VALUE RANGES FOR THE A3 BIAS VOLTAGE HISTOGRAM
47-53	IIN(47)-IIN(53)	7	SPI	PCM COUNTS	ASSIGNED VALUE RANGES FOR THE A5 BIAS VOLTAGE HISTOGRAM
54-60	IIN(54)-IIN(60)	7	SPI	PCM COUNTS	ASSIGNED VALUE RANGES FOR THE A6 BIAS VOLTAGE HISTOGRAM
61	IIN(61)	1	SPI	PCM COUNTS	LOW PCM COUNT LIMIT OF A4 FOR WAVELENGTH CAL PARABOLA MATRIX
62	IIN(62)	1	SPI	PCM COUNTS	HIGH PCM COUNT LIMIT OF A4 FOR WAVELENGTH CAL PARABOLA MATRIX
63-64	IIN(63)-IIN(64)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR ZERO VOLTS REF. (D007)

Table 2.7-13 Layout of Named Common INPUT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
65-66	IIN(65) - IIN(66)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR POWER SUPPLY DIAG. (A016)
67-68	IIN(67) - IIN(68)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR PACKAGE TEMP. (A018)
69-70	IIN(69) - IIN(70)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR DATA PALLET TEMP. (A007)
71-72	IIN(71) - IIN(72)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR SPECTROMETER PALLET TEMP (A008)
73-74	IIN(73) - IIN(74)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR DICHROIC TEMP. (A019)
75-76	IIN(75) - IIN(76)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR INT. SPHERE TEMP. (A015)
77-78	IIN(77) - IIN(78)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR MIRROR TEMP. (A009)
79-80	IIN(79) - IIN(80)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR HEATED CAL TEMP. (A013)
81-82	IIN(81) - IIN(82)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR AMBIENT CAL TEMP. (A014)
83-84	IIN(83) - IIN(84)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR LWL DETECTOR TEMP. (A020)
85-86	IIN(85) - IIN(86)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR THERMAL REF. TEMP. (A017)
87-88	IIN(87) - IIN(88)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR SWL CAL LAMP (A023)
89-90	IIN(89) - IIN(90)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR RAD CAL WHEEL POS. DURING FIELD DATA
91-92	IIN(91) - IIN(92)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR RAD CAL WHEEL POS. DURING HEATED CAL
93-94	IIN(93) - IIN(94)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR RAD CAL WHEEL POS. DURING SWL CAL
95-96	IIN(95) - IIN(96)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR RAD CAL WHEEL POS. DURING AMBIENT
					CAL
97-98	IIN(97) - IIN(98)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR FOV FLAG WHEN BOTH SWL AND LWL
					CHANNELS ARE ACTIVE
99-100	IIN(99) - IIN(100)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR FOV FLAG WHEN THE SWL CHANNELS IS
					ACTIVE
101-102	IIN(101) - IIN(102)	2	SPI	PCM COUNTS	MAXIMUM AND MINIMUM LIMITS FOR FOV FLAG WHEN THE LWL CHANN[illegible]S
					ACTIVE.

Table 2.7-13 Layout of Named Common INPUT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
103-104	RIN(1)	1	SPF		WAVELENGTH VALUE TO BE USED IN RESPONSIVITY COMPUTATION FOR
					CHANNEL A2
105-106	RIN(2)	1	SPF		CAL SOURCE BRIGHTNESS FOR CHANNEL A2
107-108	RIN(3)	1	SPF		CHANNEL BIAS VOLTAGE FOR CHANNEL A2
109-116	RIN(4) - RIN(7)	4	SPF		RESPONSIVITY POLYNOMIAL COEFFICIENT VALUES FOR CHANNEL A2
117-118	RIN(8)	1	SPF		WAVELENGTH VALUE TO BE USED IN RESPONSIVITY COMPUTATION FOR
					CHANNEL A3
119-120	RIN(9)	1	SPF		CAL SOURCE BRIGHTNESS FOR CHANNEL A3
121-122	RIN(10)	1	SPF		CHANNEL BIAS VOLTAGE FOR CHANNEL A3
123-130	RIN(11) - RIN(14)	4	SPF		RESPONSIVITY POLYNOMIAL COEFFICIENT VALUES FOR CHANNEL A3
131-132	RIN(15)	1	SPF		WAVELENGTH VALUE TO BE USED IN RESPONSIVITY COMPUTATION FOR
					CHANNEL A5
133-134	RIN(16)	1	SPF		CAL SOURCE BRIGHTNESS FOR CHANNEL A5
135-136	RIN(17)	1	SPF		CHANNEL BIAS VOLTAGE FOR CHANNEL A5
137-144	RIN(18) - RIN(21)	3	SPF		RESPONSIVITY POLYNOMIAL COEFFICIENT VALUES FOR CHANNEL A5
145-146	RIN(22)	1	SPF		WAVELENGTH VALUE TO BE USED IN RESPONSIVITY COMPUTATION FOR CH. A6
147-148	RIN(23)	1	SPF		DICHROIC REFLECTIVITY FOR CHANNEL A6
149-150	RIN(24)	1	SPF		CHANNEL BIAS VOLTAGE FOR CHANNEL A6
151-158	RIN(25) - RIN(28)	4	SPF		RESPONSIVITY POLYNOMIAL COEFFICIENT VALUES FOR CHANNEL A6

JSC-10140

Table 2.7-14 Layout of Named Common INTNDX

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	L	1	SPI		NUMERICAL INDEX FOR CAL PERIODS
2	LBIAS	1	SPI		NUMERICAL INDEX FOR BIAS VOLTAGE CAL PERIODS
3	LHEAT	1	SPI		NUMERICAL INDEX FOR HEATED CAL PERIODS
4	LSWL	1	SPI		NUMERICAL INDEX FOR SWL CAL PERIODS
5	LWLW	1	SPI		NUMERICAL INDEX FOR W/L CAL PERIODS

Table 2.7-15 Layout of Named Common LASFRM

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	LSTTME	1	DPR	SECONDS	LAST FRAME TIME FOR QA SUMMARY LISTING

Table 2.7-16 Layout of Named Common PRNDX

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-18	PRNDX1	2,9	SPI		NUMERICAL PRODUCTION INDEXES FOR PROCESSING OF DATA RECORDED AT ONE SAMPLE PER FRAME. THE FIRST SUBSCRIPT INDICATES START (1,) OR STOP (2,) INDEX. THE SECOND SUBSCRIPT INDICATES THE PROCESSING PERIOD, AS FOLLOWS:
					(,1) = CAL PERIOD DATA
					(,2) = WAVELENGTH CAL PERIOD (FIELD DATA)
					(,3) = BIAS VOLTAGE CAL PERIOD
					(,4) = HEATED CAL FIELD DATA
					(,5) = SWL CAL FIELD DATA
					(,6) = AMBIENT CAL FIELD DATA
					(,7) = HEATED CAL RESPONSIVITY DATA
					(,8) = SWL CAL RESPONSIVITY DATA
					(,9) = WAVELENGTH CAL CALCULATION DATA
19-36	PRNDX2	2,9	SPI		NUMERICAL PRODUCTION INDEXES FOR PROCESSING OF DATA RECORDED AT TWO SAMPLES PER FRAME. (SAME SUBSCRIPT SCHEME AS PRNDX1).
37-54	PRNDX3	2,9	SPI		NUMERICAL PRODUCTION INDEXES FOR PROCESSING OF DATA RECORDED AT 32 SAMPLES PER FRAME. (SAME SUBSCRIPT SCHEME AS PRNDX1).

Table 2.7-17 Layout of Named Common QADAT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1 -4	SSTME	1	DPR	SECONDS	FIRST FRAME TIME FOR QA SUMMARY PRINTOUT
5	MFLG	1	SPI		FIRST FRAME TIME STORED FLAG

Table 2.7-18 Layout of Named Common RDARG

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	INIT	1	SPI		RDARG FORMS PART OF THE INTERFACE WITH THE CONTROL CARD INPUT
2	I1	1	SPI		ROUTINE CONINP, AND IS DEFINED IN SECTION 1.2.4.2-A OF DOCUMENT
3	I2	1	SPI		NUMBER ERS-300-01, "EARTH RESOURCES INPUT PROCESSOR, INPUT
4	IPRINT	1	SPI		PROCESSOR, TABULATION INPUT PROCESSOR, AND PLOT INPUT PROCESSOR
5	NI	1	SPI		PROGRAM DOCUMENT".
6	NR	1	SPI		

Table 2.7-19 Layout of Named Common RDCNTL

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	NLIN	1	SPI		NAMED COMMON RDCNTL FORMS PART OF THE INTERFACE WITH THE CONTROL CARD INPUT ROUTINE CONINP AND IS DEFINED IN SECTION 1.2.4.2-B OF DOCUMENT NUMBER ERS-300-01, "EARTH RESOURCES INPUT PROCESSOR, TABULATION INPUT PROCESSOR, AND PLOT INPUT PROCESSOR - PROGRAM DOCUMENT".
2	NRIN	1	SPI		
3	NTITLE	1	SPI		
4	NREAD	1	SPI		
5	NC	1	SPI		
6	NB	1	SPI		
7	NLI	1	SPI		
8	NLR	1	SPI		
9-164	IFMT	12,13	SPI		
165-524	LABELI	5,72	FA		
525-564	LABELR	5,8	FA		
565-780	BND	2,54	SPR		

Table 2.7-20 Layout of Named Common RECPTR

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	IVAR2	1	SPI		DUMMY POSITION FOR STORING DEFINE FILE POINTER

Table 2.7-21 Layout of Named Common RESDAT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-2	JCMPLT	2	SPI		NUMERICAL INDEX AND COMPLETION FLAG FOR SWL CAL DATA
3-4	KCMPLT	2	SPI		NUMERICAL INDEX AND COMPLETION FLAG FOR HEATED CAL DATA
5-689	NA4	685	SPI		STORAGE ARRAY FOR TRANSFER OF CHANNEL DATA TO DISK
690	NFSCN	1	SPI		FIRST SCAN FLAG
691	NX	1	SPI		MEAS ARRAY SAMPLE COUNTER
692	NSAMPL	1	SPI		MEAS ARRAY FIRST SAMPLE POINTER
693	NIX	1	SPI		MEAS ARRAY INTERIM SAMPLE POINTER
694-695	AVJ2	1	SPR	VOLTS	AVERAGE CH. A2 VALUE ACCUMULATOR
696-697	AVJ3	1	SPR	VOLTS	AVERAGE CH. A3 VALUE ACCUMULATOR
698-699	AVJ5	1	SPR	VOLTS	AVERAGE CH. A5 VALUE ACCUMULATOR
700-701	AVJ6	1	SPR	VOLTS	AVERAGE CH. A6 VALUE ACCUMULATOR
702-703	SVJ2	1	SPR	VOLTS	ACCUMULATOR FOR CH. A2 VALUES SQUARED
704-705	SVJ3	1	SPR	VOLTS	ACCUMULATOR FOR CH. A3 VALUES SQUARED
706-707	SVJ5	1	SPR	VOLTS	ACCUMULATOR FOR CH. A5 VALUES SQUARED
708-709	SVJ6	1	SPR	VOLTS	ACCUMULATOR FOR CH. A6 VALUES SQUARED
710	JSCN6	1	SPI		CH. A6 SCAN COUNTER FOR HEATED CAL PROCESSING
711	NSSCAN	1	SPI		SCAN COUNTER FOR SWL CAL PROCESSING

Table 2.7-22 Layout of Named Common SAVE

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	IBGNSC	1	SPI		0-SEARCHING FOR START OF SCAN LINE
					1-SEARCHING FOR END OF SCAN LINE
2	NMVALD	1	SPI		NUMBER OF CONSECUTIVE VALID SCAN LINES THAT HAVE BEEN PROCESSED
3	IAHEAD	1	SPI		0-NORMAL PROCESSING
					1-BEGINNING OF SCAN LINE OCCURRED ON THE LAST SAMPLE OF THE
					PREVIOUS MEAS ARRAY
4	IPRVOK	1	SPI		0-PREVIOUS SCAN WAS NOT VALID
					1-PREVIOUS SCAN WAS VALID
5	ISAVPT	1	SPI		THE LAST POSITION IN THE MEAS ARRAY THAT CONTAINED DATA FOR THE
					PREVIOUS SCAN
6	IA4NDX	1	SPI		THE LAST POSITION FILLED IN THE IA4 ARRAY
7-10	TMCURR	1	DPR	SECONDS	END TIME FOR SCAN LINE
11-14	TMFRCR	1	DPR	SECONDS	START TIME FOR CURRENT MEAS ARRAY BEING PROCESSED
15-18	TMFRST	1	DPR	SECONDS	START TIME FOR SCAN LINE
19-22	SUMTM	1	DPR	SECONDS	OVERALL TOTAL TIME OF THE SCAN LINES PROCESSED
23-24	RMSTOT	1	SPR	PCM COUNTS	OVERALL TOTAL OF THE ROOT MEAN SQUARES OF THE SCAN LINES PROCESSED
25	IA6SET	1	SPI		0-NO A6 SYNC PULSES PROCESSED
					1-1 A6 SUNC PULSE HAS BEEN PROCESSED
					2-2 A6 SYNC PULSES HAVE BEEN PROCESSED
26-27	RMNSUM	1	SPR	PCM COUNTS	OVERALL TOTAL OF THE MINIMUM A4 VALUES FOR ALL SCAN LINES
					PROCESSED

JSC-10140

Table 2.7 22 Layout of Named Common SAVE

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
28 -29	RMXSUM	1	SPR	PCM COUNTS	OVERALL TOTAL OF THE MAXIMUM A4 VALUES FOR ALL THE SCAN LINES
					PROCESSED
30 -31	RXSMSQ	1	SPR	PCM COUNTS	OVERALL TOTAL OF THE SUM OF THE MAXIMUM A4 VALUES SQUARED
32 -33	RNSMSQ	1	SPR	PCM COUNTS	OVERALL TOTAL OF THE SUM OF THE MAXIMUM A4 VALUES SQUARED

JSC-10140

Table 2.7-23 Layout of Named Common SIXSV for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1	ISIX	1	SPI		SYNC PULSE FLAG FOR CHANNEL A6 DATA

Table 2.7-24 Layout of Named Common TIMES

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-2	BJD	1	SPR	SECONDS	THE BASE JULIAN DATE ASSOCIATED WITH THE DATE OF DATA RECORDING.
					THE BASE DATE IS LIMITED TO THE Ø HOUR OF THE DAY BECAUSE OF THE
					LIMITED ACCURACY OF SINGLE PRECISION FLOATING POINT REPRESENTATION.
					THE JULIAN DATE IS MODIFIED FOR STORAGE. THE TWO HIGH ORDER BITS,
					WHICH ARE ALWAYS 24 ARE IMPLIED BUT NOT STORED IN BJD.
3-522	SSTIME	2,65	DPF	SECONDS	START AND STOP TIMES FOR THE PERIODS TO BE PROCESSED. ASSIGNED
					POSITIONS FOR THE VARIOUS PERIODS ARE:
					(1,1) THRU (2,20) = OVERALL TIME PERIODS
					(1,21) THRU (2,25) = CAL PERIODS
					(1,26) THRU (2,30) = W/L CAL PERIODS (FIELD DATA)
					(1,31) THRU (2,35) = BIAS VOLTAGE CAL PERIODS
					(1,36) THRU (2,40) = HEATED CAL PERIODS (FIELD DATA)
					(1,41) THRU (2,45) = SWL CAL PERIODS (FIELD DATA)
					(1,46) THRU (2,50) = AMBIENT CAL PERIODS (FIELD DATA)
					(1,51) THRU (2,55) = HEATED CAL (RESPONSIVITY)
					(1,56) THRU (2,60) = SWL CAL (RESPONSIVITY)
					(1,61) THRU (2,65) = W/L CAL PROCESSING

Table 2.7-25 Layout of Named Common TITLES

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1-2	ITITLE(1)	2	FA		PROJECT IDENTIFICATION
	ITITLE(2)				
3-32	ITITLE(3)	30	FA		TITLE INFORMATION FOR TABS

Table 2.7-26 Layout of Named Common WVLDAT

for QA191H

Integer Word Number In Common	Program Symbol	Dimension	Type	Units	Description
1 -2	MCMPLT	2	SPI		NUMERICAL INDEX AND COMPLETION FLAG FOR W/L CAL PERIOD
					PROCESSING
3	LMTLO	1	SPI	PCM	LOW LIMIT PCM VALUE FOR CH. A4 ARRAY
4	LMTHI	1	SPI	PCM	HIGH LIMIT PCM VALUE FOR CH. A4 ARRAY

2.8 SUBROUTINES

Refer to Figure 2.8-1 for a cross reference between the subroutine names and the load modules which reference them. Following the figure is an explanation of the function and linkages of each of the subroutines contained in the QA191H Program.

Subroutines	Load Modules	Resident QA191H.LDA	ANPRO .LLL	BIAPRO.KKK	CONDRV.FFF	DCMDRV.III	DECRIP.RRR	ERRDRV.HHH	FLDPRO.JJJ	QAEXEC.GGG	QASUM.QQQ	RAMPRO.NNN	RESPRO.OOO	TMLOOP.PPP	WVLPRO.MMM	Data Areas DCMBUF[illegible]	
MAIN DRIVERS																	
QA191H.OBJ		X															
ANPRO .OBJ			X														
BIAPRO.OBJ				X													
CONDRV.OBJ					X												
DCMDRV.OBJ						X											
DECRIP.OBJ							X										
ERRDRV.OBJ								X									
FLDPRO.OBJ									X								
QAEXEC.OBJ										X							
QASUM.OBJ											X						
RAMPRO.OBJ												X					
RESPRO.OBJ													X				
TMLOOP.OBJ														X			
WVLPRO.OBJ															X		
BLOCK DATAS																	
BLKDAT.OBJ					X												
DATDCM.OBJ							X										
DCMBUF.OBJ						X											

Figure 2.8-1 Cross Reference of Subroutine Names with Load Modules Requiring the Subroutine

2.8.1 CONDRV

A. Description. Subroutine CONDRV reads the control input data from lead cards and stores it in specified common arrays. Lead cards are read into common arrays by support software subrcutine CONINP.

B. Linkages

1. Input

- Lead cards

2. Output

- Control input data in specified common arrays

2.8.2 DECRIP

A. Description. Subroutine DECRIP calls support software subroutine DCRIPT to read the header block from the preprocessor 9-track tape and store data in a common array.

B. Linkages

1. Input

- Array of measurement ID's for FSS
- Specified common arrays

2. Output

- Specified common arrays
- Error code

2.8.3 DECOM2

A. Description. Subroutine DECOM2 calls support software subroutine DCOM2N to read data from the preprocessor 9-track tape and store data in specified common arrays.

B. Linkages

1. Input

- Start and stop times for extracting data from input tape
- Array of measurement ID's for FSS
- Maximum number of frames of data to retrieve
- Specified common arrays

2. Output

- Actual number of frames of data retrieved
- Specified data array
- Error code

2.8.4 FLDPRO

A. Description. Subroutine FLDPRO tests specific parameters in the data input by DCOM2N. The data is tested against high and low tolerance limits input by CONINP. If anomalies are detected by the tests, the data parameter is passed to ANPRO for output. The process is as follows:

1. The parameter tolerance limits input by CONINP are initiated. For those parameters requiring more than one set of limits, the correct set is determined by the type of data being tested. Indexing flags for the various types of data are set by TMLOOP.

2. Each data sample of the housekeeping parameters is checked for values within their respective limits. If a data sample exceeds limits, the parameter identity and value are passed to ANPRO for processing.

3. During calibration data processing, the sync pulses are checked at the start of each scan. A valid sync pulse occurs when the values of all channels, except channel A4, exceed 1020 counts at a common data point during one to three consecutive samples. The data on any bad or missing sync pulse is passed to ANPRO.

4. The program start/stop times that are entered for the various processing periods are taken directly from the 14-track flight log, and the log times frequently do not exactly match the actual tape times. Consequently, there may be a few very short periods of tape data that are incorrectly identified to the program. When this occurs, the program may indicate anomalous data at the very beginning or ending of these periods. This fact should be taken in consideration when checking the real-time anomalies list.

B. Linkages

1. Input

- Control input in specified common arrays.
- Data in labeled common array DCDATA.

2. Output

- Anomalous data points passed to ANPRO.

2.8.5 ANPRO

A. Description. Subroutine ANPRO converts anomalous parameter data to engineering units and outputs the data to the line printer. When anomaly data for a specific parameter is initially passed to ANPRO, an internal flag is set for that parameter. ANPRO processing keys on these flags when subsequent anomalies of the same parameter are received. If a parameter flag has been set by previous data, subsequent anomalies of that parameter are counted for a number of specified scans (nominally 30 scans), or when time continuity is broken by more than 30 seconds. When the specified number of scans have been processed, the parameter label, the total of all anomalies detected on that parameter, and the number of scans are output to the line printer.

If the processing flag has not been set, the anomaly data is converted to engineering values of voltage or temperature, depending on the parameter type, and output to the line printer.

One constant is used to convert all applicable PCM values to volts.

VAN = NPCM * .005 volts

Where:

NPCM = Parameter value in PCM counts.

VAN = Parameter value in volts.

All temperature parameter values are converted using one generalized 6th order polynomial equation described below. Temperature conversion coefficients are read from the common table CNVTMP. Each temperature parameter has a different set of conversion coefficients.

$$TAN = A_0 + A_1C + A_2C^2 \ . \ . \ . + A_NC^N \quad N \leq 6$$

Where:

TAN = Parameter value in degrees.

$A_0, A_1, \ldots A_N$ = Conversion coefficients.

C = Parameter value in PCM counts.

B. Linkages

1. Input

- Anomalous parameter data transferred by subroutine FLDPRO.

2. Output

- Converted values, anomaly totals, data frame times, and parameter labels output to line printer in formatted form.

2.8.6 BIAPRO

A. Description. Subroutine BIAPRO generates histograms of the parameter data values during bias voltage processing. Arrays are initiated for channel A1, A2, A3, A5 and A6 with assigned values specified by table IBVHR. The values for table IBVHR are input via control cards through subroutine CONDRV. Each data sample of the five channels, except the sync pulses and the four adjoining samples on each side, is examined. The histogram counters are incremented when a data sample falls within its range. At the end of bias voltage processing, the histogram arrays are stored on disk until the new Historical File tape is written.

B. Linkages

1. Input

- Specified arrays of data stored in common.

2. Output

- Histogram data written on intermediate disk file.

2.8.7 WVLPRO

A. Description. Subroutine WVLPRO processes the Wavelength Calibration period data for the Historical File. Data is input by specific common blocks. For the first twenty valid scans of data, the channel A4 values are examined for a predetermined low and high PCM count range. Arrays are initiated, and a matrix is built of all channel A2, A3, A4 and A5 PCM values within the specified A4 range. Initially, all the values in the channel A3 array are checked for saturation, on each scan of data. On the first scan, any value that exceeds 949 PCM counts is considered saturated, and on all subsequent scans, the data is considered saturated if any value exceeds 999 PCM counts. If no saturated data is found in the channel A3 arrays during the first twenty consecutive valid scans, the wavelength cal data is derived from the computed A3 values. If saturated data is found in any of the channel A3 arrays before twenty valid scans are processed, all the channel A3 data is rejected, the intermediate accumulators are cleared, and the processor begins looking at the channel A5 data. Again, for each scan of data, all the channel A5 array values are checked for saturation, using the 949 count limit for the first scan and the 999 count limit for all subsequent scans. Processing continues on the channel A5 data, until twenty new consecutive valid scans have been processed. The processor then uses the A5 values computed from the twenty new scans to derive the wavelength cal data. However, if saturated data is found in any A5 array before enough scans have been processed, the processor rejects all channel A5 data, clears the intermediate accumulators, and beings initial processing again, using the channel

2.8.7 WVLPRO (CONT.)

A. A2 data. The channel A2 arrays are tested for saturated data, as were the channel A3 and channel A5 arrays, using the same 949 and 999 count criteria. Processing continues until twenty more valid scans of channel A2 data are processed. As before, if the A2 arrays contain no saturated data for the twenty scans, the computed A2 values are used to derive wavelength cal data. But, if saturated data is found in any channel A2 array, or if the wavelength cal period terminates before twenty valid scans of data are processed from any of the three channels, the wavelength cal processing is aborted. Applicable messages are input to the Historical File to tell the user why the wavelength cal processing was aborted or to indicate which channel data was used for the derivation.

The maximum PCM value in the chosen array, and the first and last time this value occurs in the array, is determined. A low and high point below and above the first and last maximum count is established by testing for the first and last PCM value equal or greater than one-third the maximum value. If the low and high points fall on the first or last sample in the array, or if the low or high value is not found in the array, that scan is rejected. Also, the scan is rejected if the number of samples between the low and high points is less than seven or greater than twenty. A parabolic fit of the chosen array values between the low point and the high point is computed by performing a least-squares regression on the associated A4 values. The vertex of the parabola is determined, and the representative A4 value of the vertex sample number is found. This A4 value is divided by 983 counts and multiplied by the peak value of A4 for the entire scan to compensate for A4 drift.

The best-fitting parabola is given by the equation below:

$$y = a_0 + a_1x + a_2x^2$$

and the values a_0, a_1, and a_2 are computed from the following equations:

$$\sum y = a_0n + a_1\sum x + a_2\sum x^2$$
$$\sum xy = a_0\sum x + a_1\sum x^2 + a_2\sum x^3$$
$$\sum x^2y = a_0\sum x^2 + a_1\sum x^3 + a_2\sum x^4$$

where:

n = Number of samples ($6 \leq n \leq 10$).
y = Count values for the selected channel.
x = Count values of associated A4 samples.

2.8.7 WVLPRO (CONT.)

A. Twenty values of A4 are found (one per scan) for the chosen channel (A3, A5, or A2). The average and standard deviation for the twenty values are computed and this data is stored as part of the Historical File.

B. Linkages

1. Input

- Specified common arrays.

2. Output

- Historical File data.

2.8.8 RAMPRO

A. Description. Subroutine RAMPRO processes the wavelength ramp data during Calibration period processing. Wavelength ramp data is processed for some variable number of consecutive valid scans (≤100 scans) determined by the control card input value ICSCN. Data is input through common arrays.

For each scan, the minimum and peak PCM count value of channel A4 is determined. These A4 values are accumulated, and the average and standard deviation for the minimum and peak values is computed. The PCM values are converted to volts with the algorithm:

Volts = PCM count *.005 volts/count

The algorithm for computing average voltage ($\overline{Vx}$) is:

$$\overline{Vx} = \sum_{i=1}^{n} Vx, i/n$$

The algorithm for computing standard deviation (S) is:

$$S = \sqrt{\frac{n\sum_{i=1}^{n} V_i^2 - \left(\sum_{i=1}^{n} V_i\right)^2}{n(n-1)}}$$

Where:

n = Number of consecutive valid scans
Vi = Peak or minimum value in PCM counts

2.8.8 RAMPRO

A. Description (CONT.)

RAMPRO also checks the linearity of each scan. This is done by first computing a straight line value for the channel A4 counts using the algorithm:

$$y = b_0 + b_1 x$$

where:

$$b_1 = \frac{n\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

$$b_0 = \frac{\sum y_i}{n} - b_1 \frac{\sum x_i}{n} = \bar{y} - b_1\bar{x}$$

y = computed PCM value for straight line

x = sample count (starting with first sample where A4 PCM value ≥ 30 counts)

The actual PCM value of each A4 sample is compared to the computed straight line value, and the RMS deviation of the total scan is computed by algorithm:

$$RMS = \left[\frac{1}{n}\sum_{i=1}^{n} (y_i - y_e)^2\right]^{\frac{1}{2}}$$

Where:

y_e = Expected PCM value as computed for straight line.

y_i = Actual PCM value.

n = Number of samples.

2.8.8 RAMPRO

A. Description (CONT.)

During the sample comparison, if an actual PCM value exceeds the computed straight line value by plus or minus the tolerance limit dictated by the variable ILTOL, the scan is rejected; and the frame time, A4 value in volts, and identifying label are output to the line printer.

The average length of all the scans, in seconds, is computed by accumulating the total overall time and dividing by the number of valid scans.

B. Linkages

1. Input

- o Data in common arrays
- o Control parameters from lead cards

2. Outputs

- o Data stored as part of Historical File

2.8.9 RESPRO

A. Description. Subroutine RESPRO computes the responsivity, noise, and NESR data during the Calibration period data processing. The computation is done for one wavelength in each of channels A2, A3, A5 and A6, using the data from twenty valid scans. Channels A2, A3, and A5 are done during the SWL cal period, and channel A6 is done during the heated cal period. The data is input through common arrays.

1. The input wavelength (WVLNGH) for each channel is first converted to a PCM count value representing the associated channel A4 ramp position. Conversion is done using a general third power polynomial, with the conversion coefficients provided by input control cards (ACOEFZ thru ACOEF3).

$$WVPCM = A_0 + A_1C + A_2C^2 + A_3C^3$$

Where:

WVPCM = Wavelength in PCM counts.

$A_0, \ldots, A_3$ = Conversion coefficients.

C = Wavelengths in microns.

2. Compensate for channel A4 drift by: $CWVPCM = WVPCM * \frac{PKA4}{983}$

CWVPCM = Corrected value of wavelength in counts.

PKA4 = Peak value of A4 for the given scan.

3. Obtain the detector output volts (VJ) in each of the twenty scans by determining the sample number of actual A4 values, where A4(K) is the first value that equals or exceeds CWVPCM.

$$VJ = \frac{1}{5} \sum_{i=K-2}^{K+2} Vi$$

V_i = Detector PCM value.

K = First sample number where A4 $\geq$ CWVPCM

4. After the twenty values of VJ are found, their count values are converted to volts by:

VVJ = VJ *.005 volts/count.

5. Obtain the average (VVA) and standard deviation (VVN) of VVJ.

2.8.9 RESPRO

A. 6. Obtain the signal volts (SV) by subtracting the channel bias voltage, for that particular channel, from the average (VVA). The channel bias voltage for channels A2, A3, A5, and A6 are input via control cards (CBIASV).

7. For channels A2, A3, and A5, the Responsivity is computed by:

$$\text{Responsivity} = \frac{SV}{L}$$

where:

L = Cal source brightness for the given wavelength, input by control cards (PDORL).

8. For channel A6 responsivity is computed by:
Obtaining the blackbody radiance (BWLT) for the Reference Cal Source temperature (TREF), Heated Cal Source temperature (THEAT), and Dichroic temperature (TDICH).

$$BWLT = \frac{11909}{\lambda^5 * (e^x - 1)}$$

where:

λ = Wavelength in microns.

T = Temperature in ^{o}C.

$$x = \frac{14388}{\lambda * (T+273)}$$

Calculate the radiance of the Reference Source and the Heated Source by:

$$I_r = BWLT\ (TREF)$$

$$I_H = Pd*BWLT(THEAT) + (1-Pd)*BWLT(TDICH)$$

where:

Pd = Dichroic reflectivity for channel A6 wavelength, input by control cards (PDORL).

Calculate A6 Responsivity by:

$$\text{Responsivity} = \frac{SV}{I_H - I_r}$$

2.8.9 RESPRO

A. 9. Obtain NESR for each channel by:

$$NESR = \frac{VVN}{Responsivity}$$

Where:

VVN = Computed noise for that channel.

B. Linkages

1. Input

- Data in common arrays
- Conversion coefficients, wavelengths, cal source brightness, and channel bias voltages in commons.

2. Outputs

- Data stored as part of Historical File.

2.8.10 TMLOOP

A. Description. Subroutine TMLOOP is called to set up the production index arrays in the named common PRNDX. These indices are used by QAEXEC to direct the various calculations and parameter checks on the incoming tape data. TMLOOP also generates a numerical pointer that indicates which period, within a group, is being processed. Subroutine TMLOOP extracts the first frame time from the MEAS array input data, then uses the frame delta time to calculate a last data frame time in the array. The times for the various processing periods were put in the SSTIME array by the card reader routine CONINP, and these times are compared against the first and last MEAS array frame times. A start-stop index is set in the applicable PRNDX array for the first and last data sample of each period that falls within the first and last frame times. A separate index array is used for data recorded at one sample per frame (PRNDX1), two samples per frame (PRNDX2), and 32 samples per frame (PRNDX3). Numerical pointers are assigned to the separate Cal periods, Wavelength Cal, Bias Voltage Cal, Heated Cal, and SWL Cal periods as the data for each time slice is processed. These pointers are used in several routines to determine when a different processing period has begun.

B. Linkages

1. Input

o Specific time periods.

o Data in common arrays.

2. Output

o Production index arrays.

o Numerical pointers.

2.9 LOCATION OF SOURCE, OBJECT, AND LOAD MODULE FILES

The source files for QA191H are located on cards in room 201 of the 1100 NASA 1 Building. The cards are located in a file cabinet in a drawer entitled "QA191H PROGRAM DECKS". The source files are also located, along with the object files on the "QA191H DEV" disk in room 303, Building 30 at JSC. The load modules files are located on the "QA191H PROD" production disk in Building 30 at JSC under UIC (200,200).

2.10 PROGRAM LISTING

```
$JOB COMPLE[300,006]
DATE:-27-JUL-76
TIME:-17:24:33
$RUN FORTRN
FORTRAN  V004A
*DK1: ANPRO.OBJ,LP:<DK1: ANPRO.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                    17:24:48    27-JUL-76    PAGE    1
        C     ******************************************************************
        C     *                                                                *
        C     *    LOAD MODULE: ANPRO                                          *
        C     *                                                                *
        C     ******************************************************************
        C
              DIMENSION      ANCNTR(15),         ANSTAT(15),          ANREC(15),
             *               PARLAB(54),         RPAR(18),            RFMT(15),
             *               RINT(2),            RFLT(2),             RBLK(2),
             *               TANCTR(18),         ITIM(2),             RMON(12)
        C
        C
              DOUBLE PRECISION  PARLAB,          START,               STOP,
             *                  TOL,             FRTIME,              FIRSTR,
             *                  DLT,             SSTIME
        C
        C
              INTEGER        ANCNTR,             ANSTAT,              ANREC,
             *               FMT,                FRSTR(2),            FRSTP(2)
        C
              REAL CNVTMP(60)
        C
              BYTE BDATE(9)
        C
              COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
             *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
             *               IWLFG, IRAMFG,LZNE,IPG
              COMMON / DCDATA /  MEAS(3993)
              COMMON / SAVE   /  IBGNSC         ,NMVALD         ,IAHEAD        ,
             1                   IPRVOK         ,ISAVPT         ,IA4NDX        ,
             2                   TMCURR         ,TMFRCR         ,TMFRST        ,
             3                   SUMTM          ,RMSTOT         ,IA6SET        ,
             4                   RMNSUM         ,RMXSUM         ,RXSMSQ        ,
             5                   RNSMSQ
              COMMON /TIMES/ BJD, SSTIME(2,65)
              COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
              COMMON /DCARGN/ START,  STOP,      TOL,        IFLAGG,
             1                FMT,    IDKRTP,    FRSTR(2),   FRSTP(2),
             2                IDBLE,  LU,        ISIZE,      NTH,
             3                MAX,    NAV,       INIT,       IDDC,
             4                ISTAT
              COMMON /INPUT/  IIN(102),RIN(28)
              EQUIVALENCE (MO,IIN(9))
        C
              DATA  LRECD /106/
              DATA ICOL /': '/
              DATA RFMT/
             *'(1H ',', 3X',',I2,',',A2,I',',2,A2',',F7.',',4,T2',',0,I5',',    ',
             1',I2,',',T67,',',3A8,',',T92,',',    ',',    '/
              DATA RPAR/
             *',T28',',T30',',T32',',T34',',T36',',T38',',T40',',T42',',T44',
             1',T46',',T48',',T50',',T52',',T54',',T56',',T58',',T60',',T62'/
              DATA RINT /'2X,I',',6)  '/
              DATA RFLT /'F8.4',')    '/
              DATA RBLK /'2X) ','    '/
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
FORTRAN V004A                 17:24:48     27-JUL-76    PAGE     2

            DATA RMON/
            1    'JAN-',    'FEB-',    'MAR-',    'APR-',    'MAY-',    'JUN-',
            2    'JUL-',    'AUG-',    'SEP-',    'OCT-',    'NOV-',    'DEC-'/
            DATA PARLAB/
            1    'ZERO VOL',    'TS REFER',    'ENCE      ',
            2    'POWER SU',    'PPLY DIA',    'GNOSTICS',
            3    'PACKAGE ',    'TEMPERAT',    'URE       ',
            4    'DATA PAL',    'LET TEMP',    'ERATURE ',
            5    'SPECTROM',    'ETER PAL',    'LET TEMP',
            6    'DICHROIC',    ' TEMPERA',    'TURE      ',
            7    'INTEGRAT',    'ING SPHE',    'RE TEMP ',
            8    'MIRROR T',    'EMPERATU',    'RE        ',
            9    'HEATED C',    'AL TEMPE',    'RATURE   ',
            *    'AMBIENT ',    'CAL TEMP',    'ERATURE ',
            1    'RAD CAL ',    'WHEEL PO',    'SITION   ',
            2    'FIELD OF',    ' VIEW FL',    'AG        ',
            3    'LWL DETE',    'CTOR TEM',    'PERATURE',
            4    'THERMAL ',    'REF SOUR',    'CE TEMP ',
            5    'SWL CAL ',    'LAMP VOL',    'TAGE      ',
            6    'SYNC PUL',    'SE INVAL',    'ID        ',
            7    '        ',    '        ',    '          ',
            8    'SCAN REJ',    'ECTED   ',    '          '/
      C
            DATA CNVTMP/.153262E+2,.359011E-1,.648069E-5,.882920E-8,
           *.121789E-10,.0E+0,.456401E+1,.557554E-1,4*.0E+0,.437654E+1,
           *.557587E-1,4*.0E+0,.437654E+1,.557587E-1,4*.0E+0,.317003E+0,
           *.431506E-1,-.156785E-4,.198965E-7,-.117920E-10,.451193E-14,
           *-.469973E+1,.695355E-1,-.524248E-4,.928606E-7,-.853239E-10,
           *.407347E-13,-.289909E+2,.160468E+0,-.310689E-3,.516876E-6,
           *-.436491E-9,.154279E-12,.426162E+1,.520435E-1,-.237950E-4,
           *.525359E-7,-.511638E-10,.267007E-13,.432203E+1,.449168E-1,
           *-.140419E-4,.244692E-7,-.184496E-10,.929381E-14,.610345E+2,
           *.862490E-1,4*.0E+0/
      C
            IF(LZNE .GT. 0)                                        GOTO 6500
       5999 IPG=IPG+1
            CALL DATE(BDATE)
            WRITE(6,6000)IPG,IIN(10),RMON(MO),IIN(8),BDATE
      C
       6000 FORMAT(1H1,T30,'8 1 9 1 H   P R E P R O C E S S O R   T A P E   Q U A
           * L I T Y   T E S T',/,T122,'PAGE',I4,//,'  START DATE:',I3,'-',A4,I
           *2,T107,'RUN DATE: ',9A1,//,T42,'* * * * * A N O M A L I E S   L I S
           * T * * * * *',//,T41,'PARAMETER',/,T7,'FRAME TIME',T21,'RECORD',T4
           *0,'TYPE NUMBER',T74,'PARAMETER',T107,'INDICATION',/,T5,'---------
           *---- ',T21,'------',T29,'------------------------------------',T67,
           *'------------------------',T93,'------------------------------------
           *-----',/)
      C
            LZNE=11
            IF(ISIS .EQ. 1)                                        GOTO 7640
       6500 IF(LZNE .GT. 60)                                       GOTO 5999
       7000 IF(MACT .GT. 0)                                        GOTO 7100
            IF(JPAR .EQ. 0)                                        GOTO 7600
      C
            IF(ISTAT .GT. 0)                                       GOTO 7600
            IF(JPAR .GT. 15)                                       GOTO 7300
```

```
C
      IF(LAN .GT. 1)                                            GOTO 7100
C     ******************************************************************
C     * IF FIRST ANOMA_Y, SET UP RECORD LIMIT FOR COUNTER DUMP        *
C     ******************************************************************
 7005 IF(ANCNTR(JPAR) .NE. 0)                                   GOTO 7010
      ANREC(JPAR)=FRSTR(1)+LRECD
 7010 ANCNTR(JPAR)=ANCNTR(JPAR)+1
      TANCTR(JPAR)=TANCTR(JPAR)+1
      ANSTAT(JPAR)=1
      IF(ANCNTR(JPAR) .EQ. 1)                                   GOTO 7100
      IF(FRSTR(1) .LE. ANREC(JPAR))                             GOTO 7070
      IF(NSAM .GT. 6)                                           GOTO 7020
      LI=5
                                                                GOTO 7040
 7020 IF(NSAM .GT. 12)                                          GOTO 7030
      LI=4
                                                                GOTO 7040
 70"0 LI=3
 7040 IRECD=FRSTR(1)-LI
C     ******************************************************************
C     *    DETERMINE FRAME TIME (FRTIME)
C     ******************************************************************
      CALL AMOV(MEAS(1),FIRSTR,4)
      CALL AMOV(MEAS(85),DLT,4)
      IF(LI .EQ. 5) FRTIME=FIRSTR
      IF(LI .EQ. 4) FRTIME=FIRSTR+(6*DLT)
      IF(LI .EQ. 3) FRTIME=FIRSTR+(12*DLT)
      CALL STOHMS(FRTIME,ITIM,SEC)
C     ******************************************************************
C     *    WRITE ANOMALY TOTAL                                         *
C     ******************************************************************
      JB=(JPAR*3)
      JA=JB-2
      RFMT(9)=RPAR(JPAR)
      RFMT(14)=RINT(1)
      RFMT(15)=RINT(2)
      WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,ANREC(JPAR),JPAR,(PARLA
     1B(J),J=JA,JB),ANCNTR(JPAR)
      WRITE(6,7900)
      LZNE=LZNE+1
C
      ANCNTR(JPAR)=0
      ANREC(JPAR)=0
      ANSTAT(JPAR)=0
 7070 DO 7080 M=1,15
      IF(ANSTAT(M) .EQ. 0)                                      GOTO 7080
      MACT=1
                                                                GOTO 7700
 7080 CONTINUE
      MACT=0
                                                                GOTO 7700
C     ******************************************************************
C     *    LAN=1  UPDATE ANOMALY COUNTER                               *
C     ******************************************************************
 7100 IF(JPAR .EQ.0)                                            GOTO 7600
```

```
      IF(JPAR .GT. 15)                                          GOTO 7300
      IF(ANCNTR(JPAR) .EQ. 0)                                   GOTO 7005
      IF(LAN .EQ. 1)                                            GOTO 7010
      ANSTAT(JPAR)=1
      IF(JPAR .EQ. 15)                                          GOTO 7140
      LOC=((JPAR-1)*4)+1
      IOC=LOC+84
      IF(JPAR .EQ. 14)                                          GOTO 7130
      JOC=214+(MAX*JPAR)
                                                                GOTO 7150
 7130 JOC=484
                                                                GOTO 7150
 7140 LOC=81
      IOC=165
      JOC=3976
C     ****************************************************************
C     *    COMPUTE FRAME TIME OF ANOMALY                             *
C     ****************************************************************
 7150 CALL AMOV(MEAS(LOC),FRTIME,4)
      CALL AMOV(MEAS(IOC),FIRSTR,4)
      FIRSTR=FIRSTR*(NSAM-1)
      FRTIME=FRTIME+FIRSTR
      IF(NSAM .GT. 12)                                          GOTO 7160
      IF(NSAM .GT. 6)                                           GOTO 7170
      LL=5
                                                                GOTO 7180
 7160 LL=3
                                                                GOTO 7180
 7170 LL=4
 7180 IREC=FRSTR(1)-LL
      IV=JOC+NSAM-1
      IVAL=MEAS(IV)
      CALL STOHMS(FRTIME,ITIM,SEC)
      RFMT(9)=RPAR(JPAR)
      JD=3*JPAR
      JC=JD-2
      RFMT(14)=RFLT(1)
      RFMT(15)=RFLT(2)
      FVAL=IVAL
      GOTO(7190,7190,7205,7210,7215,7220,7225,7230,7235,7240,7190,7190,7
     *255,7260,7190)JPAR
C     ****************************************************************
C     *    BRANCH TO PARAMETER WRITE ROUTINE                         *
C     ****************************************************************
 7190 RVAL=IVAL*.005
                                                                GOTO 7800
 7205 RVAL=CNVTMP(43)+(CNVTMP(44)*FVAL)+(CNVTMP(45)*(FVAL**2))+(CNVTMP(4
     16)*(FVAL**3))+(CNVTMP(47)*(FVAL**4))+(CNVTMP(48)*(FVAL**5))
                                                                GOTO 7800
C
 7210 RVAL=CNVTMP(13)+(CNVTMP(14)*FVAL)+(CNVTMP(15)*(FVAL**2))+(CNVTMP(1
     16)*(FVAL**3))+(CNVTMP(17)*(FVAL**4))+(CNVTMP(18)*(FVAL**5))
                                                                GOTO 7800
C
 7215 RVAL=CNVTMP(19)+(CNVTMP(20)*FVAL)+(CNVTMP(21)*(FVAL**2))+(CNVTMP(2
     12)*(FVAL**3))+(CNVTMP(23)*(FVAL**4))+(CNVTMP(24)*(FVAL**5))
```

```
                                                            GOTO 7800
C
 7220 RVAL=CNVTMP(49)+(CNVTMP(50)*FVAL)+(CNVTMP(51)*(FVAL**2))+(CNVTMP(5
     12)*(FVAL**3))+(CNVTMP(53)*(FVAL**4))+(CNVTMP(54)*(FVAL**5))
                                                            GOTO 7800
C
 7225 RVAL=CNVTMP(31)+(CNVTMP(32)*FVAL)+(CNVTMP(33)*(FVAL**2))+(CNVTMP(3
     14)*(FVAL**3))+(CNVTMP(35)*(FVAL**4))+(CNVTMP(36)*(FVAL**5))
                                                            GOTO 7800
C
 7230 RVAL=CNVTMP(7)+(CNVTMP(8)*FVAL)+(CNVTMP(9)*(FVAL**2))+(CNVTMP(10)*
     *(FVAL**3))+(CNVTMP(11)*(FVAL**4))+(CNVTMP(12)*(FVAL**5))
                                                            GOTO 7800
C
 7235 RVAL=CNVTMP(1)+(CNVTMP(2)*FVAL)+(CNVTMP(3)*(FVAL**2))+(CNVTMP(4)*(
     1FVAL**3))+(CNVTMP(5)*(FVAL**4))+(CNVTMP(6)*(FVAL**5))
                                                            GOTO 7800
C
 7240 RVAL=CNVTMP(25)+(CNVTMP(26)*FVAL)+(CNVTMP(27)*(FVAL**2))+(CNVTMP(2
     18)*(FVAL**3))+(CNVTMP(29)*(FVAL**4))+(CNVTMP(30)*(FVAL**5))
                                                            GOTO 7800
C
 7255 RVAL=CNVTMP(55)+(CNVTMP(56)*FVAL)+(CNVTMP(57)*(FVAL**2))+(CNVTMP
     *(58)*(FVAL**3))+(CNVTMP(59)*(FVAL**4))+(CNVTMP(60)*(FVAL**5))
                                                            GOTO 7800
C
 7260 RVAL=CNVTMP(37)+(CNVTMP(38)*FVAL)+(CNVTMP(39)*(FVAL**2))+(CNVTMP(4
     10)*(FVAL**3))+(CNVTMP(41)*(FVAL**4))+(CNVTMP(42)*(FVAL**5))
                                                            GOTO 7800
C
7300  LPAR=JPAR-15
      GOTO(7310,7320,7330,7350,7370)LPAR
C
 7310 TANCTR(JPAR)=TANCTR(JPAR)+1
      IF(NSAM .GT. 576)NSAM=NSAM-576
      IF(NSAM .GT. 384)                                     GOTO 7311
      IF(NSAM .GT. 192)                                     GOTO 7312
      LL=3
                                                            GOTO 7313
 7311 LL=5
                                                            GOTO 7313
 7312 LL=4
 7313 IREC=FRSTR(1)-LL
      CALL AMOV(MEAS(77),FIRSTR,4)
      CALL AMOV(MEAS(161),DLT,4)
      FRTIME=FIRSTR+(NSAM*DLT)
      CALL STOHMS(FRTIME,ITIM,SEC)
      RFMT(9)=RPAR(16)
      RFMT(14)=RBLK(1)
      RFMT(15)=RBLK(2)
C
      WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,IREC,JPAR,(PARLAB(J),J=
     146,48)
      WRITE(6,7940)LAN
      LZNE=LZNE+1
                                                            GOTO 7700
```

```
C
 7320 IRESFG=1
      IF(LAN .GT. 0)                                   GOTO 7322
      LIL=LHEAT-1
      FRTIME=SSTIME(1,LIL)
      CALL STOHMS(FRTIME,ITIM,SEC)
      WRITE(6,7321)ITIM(1),ITIM(2),SEC
 7321 FORMAT(1H ,'HEATED RESPONSIVITY AT ',I2,':',I2,':',F7.4,' ABORTED
     1DUE TO INSUFFICIENT SCANS')
      LZNE=LZNE+1
                                                       GOTO 7700
C
 7322 LIL=LSWL-1
      FRTIME=SSTIME(1,LIL)
      CALL STOHMS(FRTIME,ITIM,SEC)
      WRITE(6,7323)ITIM(1),ITIM(2),SEC
 7323 FORMAT(1H ,'SWL RESPONSIVITY AT ',I2,':',I2,':',F7.4,' ABORTED DUE
     1TO INSUFFICIENT SCANS')
      LZNE=LZNE+1
                                                       GOTO 7700
C
 7330 TANCTR(JPAR)=TANCTR(JPAR)+1
      IF(NSAM .GT. 576)NSAM=NSAM-576
      IF(NSAM .GT. 384)                                GOTO 7331
      IF(NSAM .GT. 192)                                GOTO 7332
      LL=3
                                                       GOTO 7333
 7331 LL=5
                                                       GOTO 7333
 7332 LL=4
 7333 IREC=FRSTR(1)-LL
      CALL AMOV(MEAS(69),FIRSTR,4)
      CALL AMOV(MEAS(153),DLT,4)
      FRTIME=FIRSTR+(DLT*(NSAM-1))
      CALL STOHMS(FRTIME,ITIM,SEC)
      RFMT(9)=RPAR(18)
      IF (LAN .GT. 0 .AND. LAN .LE. 9)               GO TO 7335
      RFMT(14)=RINT(1)
      RFMT(15)=RINT(2)
      WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,IREC,JPAR,(PARLAB(J),J=
     152,54),LAN
      WRITE(6,7950)
      LZNE=LZNE+1
                                                       GOTO 7070
7335  RFMT(14)=RBLK(1)
      RFMT(15)=RBLK(2)
      WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,IREC,JPAR,(PARLAB(J),J=
     152,54)
      LZNE=LZNE+1
      GOTO(7336,7337,7338,7339,7340,7341,7342,7343,7344)LAN
 7336 WRITE(6,7955)
                                                       GOTO 7070
 7337 WRITE(6,7960)
                                                       GOTO 7070
 7338 WRITE(6,7965)
                                                       GOTO 7070
```

```
      7339 WRITE(6,7970)
                                                         GOTO 7070
      7340 WRITE(6,7975)
                                                         GOTO 7070
      7341 WRITE(6,7980)
                                                         GOTO 7070
      7342 WRITE(6,7985)
                                                         GOTO 7070
      7343 WRITE(6,7990)
                                                         GOTO 7070
      7344 TANCTR(17)=TANCTR(17)+1
           WRITE(6,7995)
                                                         GOTO 7070
      7350 IWLFG=1
           FRTIME=SSTIME(1,LWLW)
           CALL STOHMS(FRTIME,ITIM,SEC)
           WRITE(6,7355)ITIM(1),ITIM(2),SEC
      7355 FORMAT(1H ,'W/L CAL AT ',I2,':',I2,':',F7.4,' ABORTED DUE TO')
           LZNE=LZNE+1
           IF(LAN .GT. 0)                                GOTO 7360
           WRITE(6,7356)
      7356 FORMAT(1H+,T41,' ALL CHANNELS HAVE SATURATED DATA')
                                                         GOTO 7070
      7360 WRITE(6,7365)
      7365 FORMAT(1H+,T41,' INSUFFICIENT SCANS FOR COMPUTATION')
                                                         GOTO 7070
     C
      7370 IF(NMVALD .EQ. 0) IRAMFG=1
           IF(LAN .EQ. 0)                                GOTO 7500
           FRTIME=SSTIME(1,L)
           CALL STOHMS(FRTIME,ITIM,SEC)
           IF(LAN .GT. 685)                              GOTO 7380
           WRITE(6,7375)ITIM(1),ITIM(2),SEC,LAN
      7375 FORMAT(1H ,3X,I2,':',I2,':',F7.4,' W/L RAMP SCAN TOO SHORT - ',I5,
          *' SAMPLES')
           LZNE=LZNE+1
                                                         GOTO 7070
      7380 WRITE(6,7385)ITIM(1),ITIM(2),SEC,LAN
      7385 FORMAT(1H ,3X,I2,':',I2,':',F7.4,' W/L RAMP SCAN TOO LONG  - ',I5,
          *' SAMPLES')
           LZNE=LZNE+1
                                                         GOTO 7070
      7500 WRITE(6,7550)NMVALD
      7550 FORMAT(1H ,' * * * * WAVELENGTH RAMP DATA DERIVED FROM',I5,' SCANS
          * OF PREPROCESSOR TAPE DATA')
                                                         GOTO 7070
     C
      7600 DO 7650 J=1,15
           IF(ANCNTR(J) .EQ. 0)                          GOTO 7650
           IF(ISTAT .GT. 0)                              GOTO 7610
           IF((FRSTR(1)-2) .LT. ANREC(J))                GOTO 7650
      7610 CALL AMOV(MEAS(1),FRTIME,4)
           CALL STOHMS(FRTIME,ITIM,SEC)
           IREC=FRSTR(1)-4
           RFMT(9)=RPAR(J)
           JD=3*J
```

```
      JC=JD=2
      RFMT(14)=RINT(1)
      RFMT(15)=RINT(2)
      WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,IREC,J,(PARLAB(N),N=JC,
     1JD),ANCNTR(J)
      IF(ISTAT .EQ. 0)                                      GOTO 7630
      IAREC=ANREC(J)-(FRSTR(1)-4)
      IBREC=LRECD-IAREC
      RBREC=IBREC/3.5625
      IBREC=RBREC
      IF(IBREC .EQ. 0)IBREC=1
      IBREC=IABS(IBREC)
      WRITE(6,7905)IBREC
                                                            GOTO 7635
 7630 WRITE(6,7900)
 7635 ANCNTR(J)=0
      ANREC(J)=0
      ANSTAT(J)=0
      LZNE=LZNE+1
      IF(LZNE .LE. 60)                                      GOTO 7640
      ISIS=1
                                                            GOTO 5999
 7640 ISIS=0
 7650 CONTINUE
C
C
 7700 MRET=KRET+1
      GOTO(7710,7720,7730,7740,7999,7750)MRET
C
C
 7710 CALL SEG('DK1:FLDPRO.JJJ',0,0)
C
 7720 CALL SEG('DK1:RESPRO.OOO',0,0)
C
 7730 CALL SEG('DK1:RAMPRO.NNN',0,0)
C
 7740 CALL SEG('DK1:WVLPRO.MMM',0,0)
C
 7750 CALL SEG('DK1:BIAPRO.KKK',0,0)
C
 7800 WRITE(6,RFMT)ITIM(1),ICOL,ITIM(2),ICOL,SEC,IREC,JPAR,(PARLAB(J),J=
     1JC,JD),RVAL
      LZNE=LZNE+1
      GOTO(7810,7810,7820,7820,7820,7820,7820,7820,7820,7820,7810,7810,7
     *830,7820,7810)JPAR
 7810 WRITE(6,7910)                                         GOTO 7070
 7820 WRITE(6,7920)                                         GOTO 7070
 7830 WRITE(6,7930)                                         GOTO 7070
C
C     ******************************************************************
C     *     FORMAT STATEMENTS                                          *
C     ******************************************************************
C
 7900 FORMAT(1H+,T101,'ANOMALIES IN 30 SCANS')
```

```
C
 7905 FORMAT(1H+,T101,'ANOMALIES IN',I5,' SCANS')
C
 7910 FORMAT(1H+,T101,'VOLTS')
C
 7920 FORMAT(1H+,T101,'DEG. CENTIGRADE')
C
 7930 FORMAT(1H+,T101,'DEG. KELVIN')
C
 7940 FORMAT(1H+,T100,'A',I2,' SYNC VALUE TOO LOW')
C
 7950 FORMAT(1H+,T101,' SAMPLES IN A SCAN')
C
 7955 FORMAT(1H+,T101,'CH. A4 PCM ABOVE 50 CNTS')
C
 7960 FORMAT(1H+,T101,'CH. A4 PCM BELOW 956 CNTS')
C
 7965 FORMAT(1H+,T101,'W/L CAL UPPER LIMIT')
C
 7970 FORMAT(1H+,T101,'W/L CAL LOWER LIMIT')
C
 7975 FORMAT(1H+,T101,'NO W/L LOWER LMT')
C
 7980 FORMAT(1H+,T101,'NO W/L UPPER LMT')
C
 7985 FORMAT(1H+,T101,'W/L RANGE TOO LONG')
C
 7990 FORMAT(1H+,T101,'W/L RANGE TOO SHORT')
C
 7995 FORMAT(1H+,T101,'A4 ST. LINE TOL. BAD')
C
 7999 CALL SEG('DK1:QAEXEC,GGG',0,0)
      END

ROUTINES CALLED:
DATE  , AMOV  , STOHMS, IABS  , SEG

SWITCHES = /ON,/SU,/CO

BLOCK      LENGTH
MAIN.   4548    (021610)*
ANDAT   91      (000266)
DCDATA  3993    (017462)
SAVE    25      (000062)
TIMES   522     (002024)
INTNDX  5       (000012)
DCARGN  28      (000070)
INPUT   158     (000474)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES  04594 12672
EXECUTABLES   05511 11755
ASSEMBLY      03378 16805
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:BIAPRO.OBJ,LP:<DK1:BIAPRO.FTN/ON/SU/CO:99
```

FORTRAN V004A 17:27:08 27-JUL-78 PAGE 1

```
C     *****************************************************************
C     *    BIAPRO                                                     *
C     *****************************************************************
C
C
      DOUBLE PRECISION RTME,SSTIME,DSEC
      INTEGER  ANSTAT
       INTEGER ANCNTR,ANREC
      INTEGER PRNDX1,PRNDX2,PRNDX3
      DIMENSION  IBVHR(35)
      DIMENSION ITIM(2)
      COMMON /PRNDX/ PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *               IWLFG, IRAMFG,LZNE,IPG
      COMMON / DCDATA /  MEAS(3993)
      COMMON /RESDAT/ JCMPLT(2),KCMPLT(2),NA4(685),NFSCN,NX,NSAMPL,NIX
     *,AVJ2, AVJ3, AVJ5, AVJ6, SVS2, SVJ3, SVJ5, SVJ6, JSCN6, NSSCAN
     *,    IPSOK
      COMMON /INPUT/ IIN(102),RIN(28)
      COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
      COMMON  /SIXSV/  ISIX
      COMMON /TIMES/ BJD, SSTIME(2,65)
      COMMON /BIADAT/ IBHIS1(7),  IBHIS2(7),  IBHIS3(7),  IBHIS5(7),
     *                IBHIS6(7),  ISSN,  LCMPLT(2)
      COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
     *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
C
      EQUIVALENCE  (IIN(26),IBVHR(1))
C
      DIMENSION NTABUF(50)
C
      EQUIVALENCE (IIN(22),ISYNC),(IIN(23),IEND),(IIN(24),ISMIN),
     1            (IIN(25),ISMAX)
      IF(LCMPLT(1) .EQ. 0)                               GOTO 2010
      IF(LCMPLT(1) .EQ. LBIAS)                           GOTO 2020
      IF(LCMPLT(2) .EQ. 1)                               GOTO 2010
C
C     ANOMALOUS CONDITION - NOT ENOUGH BIAS CAL SCANS
C
      DSEC=SSTIME(1,LBIAS)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2005)ITIM(1),ITIM(2),SEC
 2005 FORMAT(1H ,'  BIAS VOLTAGE HISTOGRAMS AT ',I2,':',I2,':',F7.4,' AB
     *ORTED - INSUFFICIENT SCANS')
      LCMPLT(2)=1                                        GOTO 3700

C
 2010 LCMPLT(1)=LBIAS
      LCMPLT(2)=0                                        GOTO 2030
                                                         GOTO 3999
 2020 IF(LCMPLT(2) .EQ. 1)
 2030 ILO=PRNDX3(1,3)
      IHI=PRNDX3(2,3)
C
```

```
C      ** START SCAN START/STOP DETECTION **
C      ** NFSCN = 1 SAYS NOT LOOKING FOR VERY FIRST SCAN **
C
 2100 IF(NFSCN .GT. 0)                                    GOTO 2110
      NSAMPL=1
      NIX=0
C
C      ** SET UP MEAS ARRAY LOCATORS FOR ALL SIX CHANNELS **
C
 2110 ILL=ILO-1
      LOCA1=520+ILL
      LOCA2=1096+ILL
      LOCA3=1672+ILL
      LOCA4=2248+ILL
      LOCA5=2824+ILL
      LOCA6=3400+ILL
C
C      ** NX IS THE SCAN SAMPLE COUNTER **
C
 2120 NX=0
      IF(ISTA .EQ. 1)                                     GOTO 2200
C
C      ** CHECK CHANNEL A6 FOR SYNC PULSE (ISYNC) **
C
 2130 IF(MEAS(LOCA6+NX) .LT. ISYNC)GOTO 2160
      ISIX=ISIX+1
      IF(ISIX .EQ. 2)GOTO 2160
      IF(NX-1 .LT. 0)GOTO 2170
C
C      ** CHECK CHANNEL A4 FOR MINIMUM PEAK COUNT (IEND) **
C
      IF(MEAS(LOCA4+(NX-1)) .LT. IEND)                    GOTO 2700
C
C      ** CHECK SYNC PULSES ON OTHER CHANNELS **
C
 2140 IF(MEAS(LOCA1+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA2+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA3+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA5+NX) .LT. ISYNC)GOTO 2190
      GOTO 2500
C      ** CHECK FOR SCAN TOO LONG (> ISMAX) **
C
 2160 ISIX=0
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2300
      ITST=ILO+NX+NIX-1
      IF(ITST .GE. IHI)                                   GOTO 2180
 2170 NX=NX+1
      GOTO 2130
C
C
 2180 IF(NFSCN .EQ.0)                                     GOTO 3999
C
C * * * CHECK PART OF SCAN FOR HISTO VALUES * * *
C
      NRET=1
      ISS=NSAMPL+5
```

```
      IBE=NX-1
      IF(NFSCN .GT. 1) IBS=2
      IF (IIN(3) .GE. 0)                                          GOTO 2885
      INO=2180
      CALL PDUMP(INO,INO,1)
      CALL PDUMP(IBS,IBS,1)
      CALL PDUMP(IBE,IBE,1)
                                                                  GOTO 2885
 2185 NSAMPL=NSAMPL+NX
      NFSCN=NFSCN+1
      NIX=0
                                                                  GOTO 3999
C
C     ** CHECK FOR END OF MEAS ARRAY OR DATA PERIOD **
C
 2190 IF((NX+1) .GT. IHI)GOTO 2400
      II=NX+1
      GOTO 2210
 2200 II=NX
C
C     ** CHECK NEXT VALUES IN OTHER CHANNELS FOR SYNC PULSES **
C
 2210 IF(MEAS(LOCA1+II) .LT. ISYNC)GOTO 2221
      IF(MEAS(LOCA2+II) .LT. ISYNC)GOTO 2222
      IF(MEAS(LOCA3+II) .LT. ISYNC)GOTO 2223
      IF(MEAS(LOCA5+II) .LT. ISYNC)GOTO 2225
      GOTO 2500
C
C     ** ANOMALY - BAD SYNC PULSE -- SCAN NOT REJECTED **
C
 2221 LAN=1
      GOTO 2230
 2222 LAN=2
      GOTO 2230
 2223 LAN=3
      GOTO 2230
 2225 LAN=5
 2230 JPAR=16
      NSAM=NX
      KRET=5
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2500
C
C * *  PREVIOUS SCAN INVALID -- ZERO BIAS HISTOGRAM ACCUMULATORS * * *
C
 2300 CALL ZERO(IBHIS1(1),7)
      CALL ZERO(IBHIS2(1),7)
      CALL ZERO(IBHIS3(1),7)
      CALL ZERO(IBHIS5(1),7)
      CALL ZERO(IBHIS6(1),7)
      ISSN=0
      NSAMPL=1
C     ** ANOMALY -- BAD SYNC PULSE ON CH. A6 **
      NFSCN=1
      JPAR=16
      NSAM=NX
```

```
      LAN=6
      KRET=5
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2170
C
C * * END OF MEAS ARRAY DATA - CHECK DATA FOR HISTO VALUES * *
C
 2400 ISTA=1
      NRET=3
      IBS=NSAMPL
      IBE=NSAMPL+NX-5
      IF (IIN(3) .GE. 0)                                     GOTO 2885
      INO=2400
      CALL PDUMP(INO,INO,1)
      CALL PDUMP(IBS,IBS,1)
      CALL PDUMP(IBE,IBE,1)
                                                             GOTO 2885
C
C * * CHECK FOR CORRECT NUMBER OF SAMPLES IN SCAN * *
C
 2500 IF(NFSCN .EQ. 0)                                       GOTO 2530
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2510
      IF((NSAMPL+NX) .LT. ISMIN)GOTO 2510
      IPSOK=1
      GOTO 2530
C
C     ** ANOMALY -- SCAN TOO LONG OR TOO SHORT **
C
 2510 JPAR=18
      NSAM=NX
      IPSOK=0
      NFSCN=0
      LAN=NSAMPL+NX
      KRET=5
      CALL SEG('DK1:ANPRO.LLL',2)
      IHI=IHI-NIX
C
C     ** CHECK FOR FIRST CH. A4 SAMPLE LESS THAN 50 PCM CNTS **
C
 2530 II=LOCA4+NX+1
      IF(MEAS(II) .LT. 50)GOTO 2560
      NFSCN=0
C
C     ** ANOMALY -- FIRST SAMPLE IN A4 TOO HIGH **
C
      JPAR=18
      NSAM=NX
      LAN=2
      KRET=5
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2570
 2560 NFSCN=NFSCN+1
C
C     ** IS PREVIOUS SCAN OKAY? **
C
 2570 IF(IPSOK .EQ. 1)                                       GOTO 2580
```

```
      LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
      NIX=NX
      NX=1
      NSAMPL=1
                                                      GOTO 2130
C
C * *  CHECK MEAS ARRAY DATA VALUES FOR HISTOGRAM * *
C
 2580 NRET=2
      IBS=1
      IBE=NX-5
      IF (IIN(3) .GE. 0)                              GOTO 2885
      INO=2580
      CALL PDUMP(INO,INO,1)
      CALL PDUMP(IBS,IBS,1)
      CALL PDUMP(IBE,IBE,1)
                                                      GOTO 2885
 2600 ISSN=ISSN+1
      IF(ISSN .LT. 2)                                 GOTO 2650
      LCMPLT(2)=1
                                                      GOTO 3070
 2650 LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
      NIX=NX
      NX=1
      NSAMPL=1
                                                      GOTO 2130
 2700 CALL ZERO(IBHIS1(1),7)
      CALL ZERO(IBHIS2(1),7)
      CALL ZERO(IBHIS3(1),7)
      CALL ZERO(IBHIS5(1),7)
      CALL ZERO(IBHIS6(1),7)
      ISSN=0
      IPSOK=0
      NFSCN=0
      NSAMPL=1
      JPAR=18
      KRET=5
      NSAM=NX
      LAN=2
      CALL SEG('DK1:ANPRO.LLL',2)
                                                      GOTO 2140
 2885 DO 3000 L=IBS,IBE
      DO 2900 M=1,7
      IF(MEAS(LOCA1+L) .NE. IBVHR(M))                 GOTO 2900
      IBHIS1(M)=IBHIS1(M)+1
                                                      GOTO 2910
```

```
 2900 CONTINUE
      IF(MEAS(LOCA1+L) .LE. IBVHR(1))                         GOTO 2905
      IBHIS1(7)=IBHIS1(7)+1
                                                              GOTO 2910
 2905 IBHIS1(1)=IBHIS1(1)+1
C
C    * * *  CHANNEL A2
C
 2910 DO 2915 M=8,14
      IF(MEAS(LOCA2+L) .NE. IBVHR(M))                         GOTO 2915
      MM=M-7
      IBHIS2(MM)=IBHIS2(MM)+1
                                                              GOTO 2925
 2915 CONTINUE
      IF(MEAS(LOCA2+L) .LE. IBVHR(8))                         GOTO 2920
      IBHIS2(7)=IBHIS2(7)+1
                                                              GOTO 2925
 2920 IBHIS2(1)=IBHIS2(1)+1
C
C * * * CHANNEL A3
C
 2925 DO 2930 M=15,21
      IF(MEAS(LOCA3+L) .NE. IBVHR(M))                         GOTO 2930
      MM=M-14
      IBHIS3(MM)=IBHIS3(MM)+1
                                                              GOTO 2940
 2930 CONTINUE
      IF(MEAS(LOCA3+L) .LE. IBVHR(15))                        GOTO 2935
      IBHIS3(7)=IBHIS3(7)+1
                                                              GOTO 2940
 2935 IBHIS3(1)=IBHIS3(1)+1
C
C * * * CHANNEL A5
C
 2940 DO 2955 M=22,28
      IF(MEAS(LOCA5+L) .NE. IBVHR(M))                         GOTO 2955
      MM=M-21
      IBHIS5(MM)=IBHIS5(MM)+1
                                                              GOTO 2965
 2955 CONTINUE
      IF(MEAS(LOCA5+L) .LE. IBVHR(22))                        GOTO 2960
      IBHIS5(7)=IBHIS5(7)+1
                                                              GOTO 2965
 2960 IBHIS5(1)=IBHIS5(1)+1
C
C * * * CHANNEL A6
C
 2965 DO 2980 M=29,35
      IF(MEAS(LOCA6+L) .NE. IBVHR(M))                         GOTO 2980
      MM=M-28
      IBHIS6(MM)=IBHIS6(MM)+1
                                                              GOTO 3000
 2980 CONTINUE
      IF(MEAS(LOCA6+L) .LE. IBVHR(29))                        GOTO 2985
      IBHIS6(7)=IBHIS6(7)+1
                                                              GOTO 3000
```

```
FORTRAN V004A                   17:27:08     27-JUL-76     PAGE     7
     2985 IBHIS6(1)=IBHIS6(1)+1
     3000 CONTINUE
          IF (IIN(3) .GE. 0)                                    GOTO 3061
          WRITE(6,3010)(IBVHR(J),J=1,7),(IBHIS1(K),K=1,7)
     3010 FORMAT(1H ,'CHANNEL A1 BIAS VOLTAGE HISTOGRAM',/,' RANGE ',7I5,/,T
         *6,'=================================',/,T6,7I5)
          WRITE(6,3020)(IBVHR(J),J=8,14),(IBHIS2(K),K=1,7)
     3020 FORMAT(1H ,'CHANNEL A2 BIAS VOLTAGE HISTOGRAM',/,' RANGE ',7I5,/,T
         *6,'=================================',/,T6,7I5)
          WRITE(6,3030)(IBVHR(J),J=15,21),(IBHIS3(K),K=1,7)
     3030 FORMAT(1H ,'CHANNEL A3 BIAS VOLTAGE HISTOGRAM',/,' RANGE ',7I5,/,T
         *6,'=================================',/,T6,7I5)
          WRITE(6,3050)(IBVHR(J),J=22,28),(IBHIS5(K),K=1,7)
     3050 FORMAT(1H ,'CHANNEL A5 BIAS VOLTAGE HISTOGRAM',/,' RANGE ',7I5,/,T
         *6,'=================================',/,T6,7I5)
          WRITE(6,3060)(IBVHR(J),J=29,35),(IBHIS6(K),K=1,7)
     3060 FORMAT(1H ,'CHANNEL A6 BIAS VOLTAGE HISTOGRAM',/,' RANGE ',7I5,/,T
         *6,'=================================',/,T6,7I5)
     3061 CONTINUE
          GOTO(2185,2600,3999)NRET
     3070 NTABUF(1)=14
          NTABUF(2)=1
          CALL AMOV(IBHIS1(1),NTABUF(7),7)
          CALL NTRAN(12,1,50,NTABUF(1))
     3100 CALL NTRAN(12,15,LSTAT)
          IF(LSTAT+1)3110,3100,3120
     3110 LULU=12
          WRITE(6,3800)LULU,LSTAT
                                                                GOTO 4000
     3120 NTABUF(1)=15
          NTABUF(2)=2
          CALL AMOV(IBHIS2(1),NTABUF(7),7)
          CALL NTRAN(12,1,50,NTABUF(1))
     3200 CALL NTRAN(12,15,LSTAT)
          IF(LSTAT+1)3210,3200,3220
     3210 LULU=12
          WRITE(6,3800)LULU,LSTAT
                                                                GOTO 4000
     3220 NTABUF(1)=16
          NTABUF(2)=3
          CALL AMOV(IBHIS3(1),NTABUF(7),7)
          CALL NTRAN(12,1,50,NTABUF(1))
     3300 CALL NTRAN(12,15,LSTAT)
          IF(LSTAT+1)3310,3300,3320
     3310 LULU=12
          WRITE(6,3800)LULU,LSTAT
                                                                GOTO 4000
     3320 NTABUF(1)=17
          NTABUF(2)=5
          CALL AMOV(IBHIS5(1),NTABUF(7),7)
          CALL NTRAN(12,1,50,NTABUF(1))
     3500 CALL NTRAN(12,15,LSTAT)
          IF(LSTAT+1)3510,3500,3520
     3510 LULU=12
          WRITE(6,3800)LULU,LSTAT
                                                                GOTO 4000
```

```
 3520 NTABUF(1)=18
      NTABUF(2)=6
      CALL AMOV(IBHIS6(1),NTABUF(7),7)
      CALL NTRAN(12,1,50,NTABUF(1))
 3600 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3610,3600,3620
 3610 LULU=12
      WRITE(6,3800)LULU,LSTAT
                                                         GOTO 4000
 3620 IBVFLG=1
 3630 NX=0
      NIX=0
      NSAMPL=0
      NFSCN=0
      IPSOK=0
      ISSN=0
                                                         GOTO 3999
C     *******************************************************************
C     *   WRITE ZEROS IN THE HISTORICAL FILE RECORD                     *
C     *******************************************************************
 3700 NTABUF(1)=14
      CALL ZERO(NTABUF(2),49)
      CALL NTRAN(12,1,50,NTABUF(1))
 3702 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3704,3702,3706
 3704 LUL=12
      WRITE(6,3800)LUL,LSTAT
                                                         GOTO 4000
 3706 NTABUF(1)=15
      CALL NTRAN(12,1,50,NTABUF(1))
 3712 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3714,3712,3716
 3714 LUL=12
      WRITE(6,3800)LUL,LSTAT
                                                         GOTO 4000
 3716 NTABUF(1)=16
      CALL NTRAN(12,1,50,NTABUF(1))
 3722 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3724,3722,3726
 3724 LUL=12
      WRITE(6,3800)LUL,LSTAT
                                                         GOTO 4000
 3726 NTABUF(1)=17
      CALL NTRAN(12,1,50,NTABUF(1))
 3732 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3734,3732,3736
 3734 LUL=12
      WRITE(6,3800)LUL,LSTAT
                                                         GOTO 4000
 3736 NTABUF(1)=18
      CALL NTRAN(12,1,50,NTABUF(1))
 3742 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3744,3742,3746
 3744 LUL=12
      WRITE(6,3800)LUL,LSTAT
                                                         GOTO 4000
```

```
FORTRAN V004A                        17:27:08    27-JUL-78   PAGE    9

         3746 IBVFLG=1
                                                                 GOTO 3630
         3800 FORMAT(1H0,'NTRAN WRITE ERROR ON UNIT',I4,' STATUS WORD',I4)
         3999 CALL SEG('DK1:QAEXEC,GGG',0,0)
         4000 CALL SEGRET
              END

        ROUTINES CALLED:
        STOHMS, PDUMP , SEG    , ZERO   , AMOV   , NTRAN , SEGRET


        SWITCHES = /ON,/SU,/CO

        BLOCK        LENGTH
        MAIN.    3170    (014304)*
        PRNDX    54      (000154)
        ANDAT    91      (000266)
        DCDATA   3993    (017462)
        RESDAT   712     (002620)
        INPUT    158     (000474)
        INTNDX   5       (000012)
        SIXSV    1       (000002)
        TIMES    522     (002024)
        BIADAT   38      (000114)
        HISDAT   8       (000020)

        **COMPILER ----- CORE**
           PHASE        USED  FREE
        DECLARATIVES 04662 12604
        EXECUTABLES  05031 12235
        ASSEMBLY     02801 17382
```

```
*
$EOD
$RUN FORTRN
FORTRAN  V004A
*DK1:BLKDAT.OBJ,LP:<DK1:BLKDAT.FTN/ON/SU/CO:99
```

```
*
$EOD
$RUN FORTRN
FORTRAN  V004A
*DK1:BLKDAT,OBJ,LP:<DK1:BLKDAT.FTN/ON/SU/CO:99
```

```
      BLOCK DATA
C     ****************************************************************
C     *                                                              *
C     *    BLOCK DATA SUBPROGRAM FOR CONTROL INPUT PROCESSOR         *
C     *    BLOCK DATA: QA191H                                        *
C     *                                                              *
C     ****************************************************************
C
C
C     ****************************************************************
C     *    TYPE STATEMENTS                                           *
C     ****************************************************************
      DOUBLE PRECISION
     1          DLAB,
     2          SSTIME
C     ****************************************************************
C     *    DIMENSION STATEMENTS                                      *
C     ****************************************************************
      DIMENSION
     1          DLAB( 90),
     2          IBND(4,54)
C     ****************************************************************
C     *    COMMON STATEMENTS                                         *
C     ****************************************************************
      COMMON / INPUT  /  IIN(102)      ,RIN(28)
      COMMON / ERROR  /  IERR(20)
      COMMON / RDARG  /  INIT          ,I1             ,I2             ,
     1                   IPRINT        ,NI             ,NR
      COMMON / RDCNTL /  NIIN          ,NRIN           ,NTITLE         ,
     1                   NREAD         ,NC             ,NB             ,
     2                   NLI           ,NLR            ,IFMT(12,13)    ,
     3                   LABELI(5,72)  ,LABELR(5,8)    ,BND(2,54)
      COMMON / TIMES  /  BJD           ,SSTIME(2,65)
      COMMON / TITLES /  ITITL(32)
C     ****************************************************************
C     *    EQUIVALENCE STATEMENTS                                    *
C     ****************************************************************
      EQUIVALENCE        (IBND(1,1),BND(1,1))          ,(DLAB,LABELI)
C     ****************************************************************
C     *    DATA STATEMENTS                                           *
C     ****************************************************************
C     ****************************************************************
C     * FMT     INDEX     TYPE      INDEX     INDEX     INDEX        *
C     *  NO     TO NO     OF        TO        TO        INTGR        *
C     *         RECS      CONVR     BOUND     IIN       LABEL        *
C     *         TO RD               ARRAY     ARRAY     ARRAY        *
C     *     NO OF     NO OF     INCRM     INCRM     INDEX     INDEX  *
C     *     RECS      PARAM     TO        TO        TO        REAL   *
C     *     TO BE     IN        CONVR     BOUND     RIN       LABEL  *
C     *     READ      RECRD     FACTR     VALUE     ARRAY     ARRAY  *
C     ****************************************************************
      DATA IFMT/
     1  1,    1,    0,    7,    0,    0,    0,    0,    1,    0,    1,    0,
     2  7,    1,    0,   32,   -3,    0,    0,    0,    1,    0,    0,    0,
     3  1,    1,    0,    8,   -1,    0,    1,    1,    8,    0,    8,    0,
```

```
FORTRAN V004A                        17:29:24    27-JUL-76    PAGE    2
     4  1,   1,   0,   4,   0,   0,   9,   1,  16,   0,  16,   0,
     5  2,  65,  19,   6,  -2,   0,  13,   1,   0,   0,   0,   0,
     6  1,   1,   0,   6,   0,   0,  19,   1,  20,   0,  20,   0,
     7  1,   5,   0,   7,   0,   0,  25,   0,  26,   0,  26,   0,
     8  3,   4,   0,   7,   0,   0,  26,   0,   0,   1,   0,   1,
     9  1,   1,   0,   2,   0,   0,  27,   0,  61,   0,  33,   0,
     A  1,   3,   0,   8,   0,   0,  28,   0,  63,   0,  35,   0,
     B  1,   1,   0,   2,   0,   0,  28,   0,  87,   0,  35,   0,
     C  1,   1,   0,   8,   0,   0,  29,   0,  89,   0,  43,   0,
     D  1,   1,   0,   6,   0,   0,  30,   0,  97,   0,  51,   0/
      DATA NIIN/102/,NRIN/ 28/,NTITLE/ 32/,NREAD/ 13/,NC/  0/,NB/ 54/
      DATA  NLI/ 72/, NLR/  8/
      DATA
     1              IBND(1, 1) /   50    /,   IBND(3, 1) /   99   /,
     2              IBND(1, 2) /    1    /,   IBND(3, 2) /   12   /,
     3              IBND(1, 3) /    1    /,   IBND(3, 3) /   31   /,
     4              IBND(1, 4) /    0    /,   IBND(3, 4) /    3   /,
     5              IBND(1, 5) /    0    /,   IBND(3, 5) /    1   /,
     6              IBND(1, 6) /    0    /,   IBND(3, 6) / 1440   /,
     7              IBND(1, 7) /    0    /,   IBND(3, 7) /   60   /,
     8              IBND(1, 8) /    0    /,   IBND(3, 8) /   60   /,
     9              IBND(1, 9) /    1    /,   IBND(3, 9) /   20   /,
     A              IBND(1,10) /    0    /,   IBND(3,10) /    5   /,
     B              IBND(1,11) /    0    /,   IBND(3,11) /    5   /,
     C              IBND(1,12) /    1    /,   IBND(3,12) /   65   /,
     D              IBND(1,13) /    0    /,   IBND(3,13) / 1440   /,
     E              IBND(1,14) /    0    /,   IBND(3,14) /   60   /,
     F               BND(1,15) /    0.   /,    BND(2,15) /   60.  /,
     G              IBND(1,16) /    0    /,   IBND(3,16) / 1440   /,
     H              IBND(1,17) /    0    /,   IBND(3,17) /   60   /,
     I               BND(1,18) /    0.   /,    BND(2,18) /   60.  /,
     J              IBND(1,19) /    1    /,   IBND(3,19) /  100   /,
     K              IBND(1,20) /    0    /,   IBND(3,20) / 1023   /,
     L              IBND(1,21) /    0    /,   IBND(3,21) / 1023   /,
     M              IBND(1,22) /    0    /,   IBND(3,22) / 1023   /,
     N              IBND(1,23) /    0    /,   IBND(3,23) /  685   /,
     O              IBND(1,24) /    0    /,   IBND(3,24) /  685   /,
     P              IBND(1,25) /    0    /,   IBND(3,25) / 1023   /,
     Q               BND(1,26) /-1.0E38  /,    BND(2,26) / 1.0E38 /,
     R              IBND(1,27) /    0    /,   IBND(3,27) / 1023   /,
     S              IBND(1,28) /    0    /,   IBND(3,28) / 1023   /,
     T              IBND(1,29) /    0    /,   IBND(3,29) / 1023   /,
     U              IBND(1,30) /    0    /,   IBND(3,30) / 1023   /
      DATA DLAB/
     1    'SENSOR   ',  '   REC. F',  'ORM.MISS',  'ION    FL',  'IGHT NO.',
     2    'SITE NO.',   '   LINE N',  'O.   RUN ', 'NO.    YE',  'AR       ',
     3    'MONTH    ',  '   DAY    ', '      TAB ', 'OPTIONFL',  ' DEL OPT',
     4    'HRS. DEL',   '   MINS. ',  'DEL SECS',  '. DEL OV',   'RL.PERS.',
     5    'CAL PERS',   '. WVL PE',   'RS. TOTL',  ' PERS.CN',   'SEC SCNS',
     6    'LINEAR D',   'EVMNPCM ',   'SYNCA4 P',  'CM ENDMI',   'N SAMPLS',
     7    'MAX SAMP',   'LSHST RN',   'GE 1HST ',  'RNGE 2HS',   'T RNGE 3',
     8    'HST RNGE',   ' 4HST RN',   'GE 5HST ',  'RNGE 6HS',   'T RNGE 7',
     9    'LWPCM PR',   'ABHIPCM ',   'PRABMAX ',  'TOL 1 MI',   'N TOL 1 ',
     1    'MAX TOL ',   '2 MIN TO',   'L 2 MAX ',  'TOL 3 MI',   'N TOL 3 ',
     2    'MAX TOL ',   '4 MIN TO',   'L 4 MX F',  'LD RADMN',   ' FLD RAD',
     3    'MX HTD R',   'ADMN HID',   ' RADMX 5',  'WL RADMN',   ' SWL RAD',
```

```
FORTRAN V004A                    17:29:24    27-JUL-76    PAGE    3
         4    'MX AMB R',  'ADMN AMB',  ' RADMX B',  'TH FOVMN',  ' BTH FOV',
         5    'MX SWL F',  'OVMN SWL',  ' FOVMX L',  'WL FOVMN',  ' LWL FOV',
         6    '        ',  '        ',  '        ',  '        ',  '        ',
         7    '        ',  '        ',  '        ',  '        ',  '        ',
         8    '        ',  '        ',  '        ',  '        ',  '        ',
         9    '        ',  '        ',  '        ',  '        ',  '        '/
          DATA LABELR/
         1                   'WA','VE','LE','NG','TH',
         2                   'CA','L ','SR','C ','BR',
         3                   'CH','NL',' B','IV','LT',
         4                   'RS','P ','CO','EF',' 0',
         5                   'RS','P ','CO','EF',' 1',
         6                   'RS','P ','CO','EF',' 2',
         7                   'RS','P ','CO','EF',' 3',
         8                   '  ','  ','  ','  ','  '/
          END

       SWITCHES = /ON,/SU,/CO

       BLOCK        LENGTH
       DATA.    0      (000000)*
       INPUT    158    (000474)
       ERROR    20     (000050)
       RDARG    6      (000014)
       RDCNTL   780    (003030)
       TIMES    522    (002024)
       TITLES   32     (000100)

       **COMPILER ----- CORE**
          PHASE       USED  FREE
       DECLARATIVES  03750 13516
       EXECUTABLES   04165 13101
       ASSEMBLY      00941 19242
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:CONDRV.OBJ,LP:<DK1:CONDRV.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                      17:30:00     27-JUL-78    PAGE    1
        C      *********************************************************************
        C      *                                                                   *
        C      *                                                                   *
        C      *     LOAD MODULE : CONDRV                                          *
        C      *                                                                   *
        C      *                                                                   *
        C      *********************************************************************
        C
        C
        C
               DOUBLE PRECISION SSTIME
        C      *********************************************************************
        C      *     COMMON STATEMENTS                                             *
        C      *********************************************************************
               COMMON / INPUT   /  IIN(102)        ,RIN(28)
               COMMON / ERROR   /  IERR(20)
               COMMON / RDARG   /  INIT            ,I1             ,I2             ,
              1                    IPRINT          ,NI             ,NR
               COMMON / RDCNTL  /  NIIN            ,NRIN           ,NTITLE         ,
              1                    NREAD           ,NC             ,NB             ,
              2                    NLI             ,NLR            ,IFMT(12,13)    ,
              3                    LABELI(5,72)    ,LABELR(5,8)    ,BND(2,54)
               COMMON / TIMES   /  BJD             ,SSTIME(2,65)
        C
        C
               INIT     = 0
               I1       =1
               I2       =13
               IPRINT   = 3
               NI       = 0
               NR       = 0
               CALL CONINP
               IF (IERR(1) .EQ. 0) GO TO 20
               DO 10  I=1,3
               IF (IERR(I) .NE. 8 .AND. IERR(I) .NE. 9 .AND. IERR(I) .NE. 10)
              *   GO TO 15
               IERR(I) = 0
            10 CONTINUE
               I        = 4
            15 IF (IERR(I) .EQ. 0) GO TO 20
               IERR(1) =69
               IERR(2) = 0
               CALL SEG('DK1:ERRDRV,HHH',2)
            20 CONTINUE
               CALL SEGRET
               END

        ROUTINES CALLED:
        CONINP, SEG    , SEGRET

        SWITCHES = /ON,/SU,/CO

        BLOCK          LENGTH
        MAIN.     202     (000624)*
        INPUT     158     (000474)
        ERROR     20      (000050)
        RDARG     6       (000014)
        RDCNTL    780     (003030)
        TIMES     522     (002024)
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
FORTRAN V004A                    17:30:00    27-JUL-76   PAGE    2

        **COMPILER ----- CORE**
           PHASE       USED  FREE
        DECLARATIVES 03750 13516
        EXECUTABLES  04214 13052
        ASSEMBLY     01156 19027
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:DATDCM,OBJ,LP:<DK1:DATDCM,FTN/ON/SU/CO:99
```

```
FORTRAN V004A                    17:30:28    27-JUL-76    PAGE    1

C     ******************************************************************
C     *    BLOCK  DATA FOR DECOM2 FOR QA191H                           *
C     ******************************************************************
      BLOCK DATA
      DOUBLE PRECISION    TABL1
      DIMENSION TABL1(21)
      COMMON / DCCNTL /   ISENSR        ,NTPS          ,IDDEC         ,
     1                    NM            ,MIDH(84)
      EQUIVALENCE         (MIDH,         TABL1)
      DATA                ISENSR/15/,    NTPS/1/,
     1                    IDDEC / 1/,    NM  /21/
      DATA TABL1 / 'D007-RRO','A016-RRO','A018-RRO','A007-RRO',
     1             'A008-RRO','A019-RRO','A015-RRO','A009-RRO',
     2             'A013-RRO','A014-RRO','D005-RRO','D006-RRO',
     3             'A020-RRO','A017-RRO','A001-RRO','A002-RRO',
     4             'A003-RRO','A004-RRO','A005-RRO','A006-RRO',
     5             'A023-RRO'/
      END

SWITCHES = /ON,/SU,/CO

BLOCK        LENGTH
DATA.   0       (000000)*
DCCNTL  88      (000260)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES 03750 13516
EXECUTABLES  03853 13413
ASSEMBLY     00836 19347
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:DCMBUF.OBJ,LP:<DK1:DCMBUF.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                      17:30:50    27-JUL-76    PAGE    1

        C     ***************************************************************
        C     *    BLOCK DATA FOR DCOM2N                                    *
        C     ***************************************************************
               BLOCK DATA
               COMMON / DCDATA /  MEAS(3993)
               END

        SWITCHES = /ON,/SU,/CO

        BLOCK        LENGTH
        DATA.   0       (000000)*
        DCDATA  3993    (017462)

        **COMPILER ----- CORE**
           PHASE        USED  FREE
        DECLARATIVES 03750 13516
        EXECUTABLES  03839 13427
        ASSEMBLY     00800 19383
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:DCMDRV.OBJ,LP:<DK1:DCMDRV.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                    17:31:09     27-JUL-76    PAGE    1
      C*********************************************************************
      C*                                                                   *
      C*         DCOM2N DRIVER PROGRAM                                     *
      C*                                                                   *
      C*********************************************************************
      C
            DOUBLE PRECISION START, STOP, TOL, SSTIME
            INTEGER FRSTR, FRSTP, FMT
            COMMON /TIMES/ BJD, SSTIME(2,65)
            COMMON /ERROR/ IERR(20)
            COMMON / DCDATA /  MEAS(3993)
            COMMON /DCARGN/ START,  STOP,     TOL,      IFLAGG,
           1                FMT,    IDKRTP,   FRSTR(2), FRSTP(2),
           2                IDBLE,  LU,       ISIZE,    NTH,
           3                MAX,    NAV,      INIT,     IDDC,
           4                ISTAT
      C
            IERR(1)=0
            CALL DCOM2N
            IF((IERR(1).GT.0).AND.(IERR(1).LE.999))
           1 CALL SEG('ERRDRV.HHH',0,0)
            CALL SEG('DK1:QAEXEC.GGG',0,0)
            END

      ROUTINES CALLED:
      DCOM2N, SEG

      SWITCHES = /ON,/SU,/CO

      BLOCK      LENGTH
      MAIN.    96     (000300)*
      TIMES    522    (002024)
      ERROR    20     (000050)
      DCDATA   3993   (017462)
      DCARGN   28     (000070)

      **COMPILER ----- CORE**
         PHASE       USED  FREE
      DECLARATIVES  03750 13516
      EXECUTABLES   04124 13142
      ASSEMBLY      01031 19152
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:DECRIP,OBJ,LP:<DK1:DECRIP.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                    17:31:32     27-JUL-76    PAGE     1

        C*********************************************************************
        C*                                                                    *
        C*         DCRIPT DRIVER PROGRAM                                      *
        C*                                                                    *
        C*********************************************************************
        C
              DOUBLE PRECISION START, STOP, TOL, SSTIME
              INTEGER FRSTR, FRSTP, FMT
        C
              COMMON / DCCNTL /  ISENSR        ,NTPS         ,IDDEC          ,
             1                   NM            ,MIDH(84)
        C
              COMMON /DCARGN/ START,  STOP,     TOL,       IFLAGG,
             1                FMT,    IDKRTP,   FRSTR(2),  FRSTP(2),
             2                IDBLE,  LU,       ISIZE,     NTH,
             3                MAX,    NAV,      INIT,      IDDC,
             4                ISTAT
        C
              COMMON /DCDATA/MEAS(3993)
              COMMON /ERROR/ IERR(20)
              COMMON /TIMES/ BJD, SSTIME(2,65)
        C
              FMT=27
              IERR(1)=0
              CALL DCRIPT
              IF((IERR(1).GT.0).AND.(IERR(1).LE.999))
             1 CALL SEG('DK1:ERRDRV,HHH',0,0)
              CALL SEG('DK1:QAEXEC,GGG',0,0)
              END

       ROUTINES CALLED:
       DCRIPT, SEG

       SWITCHES = /ON,/SU,/CO

       BLOCK        LENGTH
       MAIN.    108     (000330)*
       DCCNTL   88      (000260)
       DCARGN   28      (000070)
       DCDATA   3993    (017462)
       ERROR    20      (000050)
       TIMES    522     (002024)

       **COMPILER ----- CORE**
          PHASE        USED  FREE
       DECLARATIVES  03750 13516
       EXECUTABLES   04137 13129
       ASSEMBLY      01072 19111
```

```
*
$EOD
$RUN FORTRN
FORTRAN  V004A
*DK1:ERRDRV.OBJ,LP:<DK1:ERRDRV.FTN/ON/SU/CO:99
```

```
FORTRAN V004A                    17:31:55    27-JUL-76    PAGE    1

C     ******************************************************************
C     *    MAIN DRIVER FOR LOAD MODULE ERRDRV                          *
C     ******************************************************************
      COMMON / ERROR  /  IERR(20)
      COMMON / ERROR1 /  N,             IUARRY(10)
      COMMON / INPUT  /  IIN(102),      RIN(28)
      CALL ERRPRC
      IF(IERR(1) .EQ. 69) CALL SEGRET
      CALL SEG('DK1:QAEXEC.GGG',0,0)
      END

ROUTINES CALLED:
ERRPRC, SEGRET, SEG

SWITCHES = /ON,/SU,/CO

BLOCK        LENGTH
MAIN.    67      (000206)*
ERROR    20      (000050)
ERROR1   11      (000026)
INPUT    158     (000474)

**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES 03750 13516
EXECUTABLES  03940 13326
ASSEMBLY     00942 19241
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:FLDPRO.OBJ,LP:<DK1:FLDPRO.FTN/ON/SU/CO:99
```

```
C*********************************************************************
C*                                                                   *
C     **** LOAD MODULE: FLDPRO ***
C*                                                                   *
C*********************************************************************
C
C
C     .......... TYPE STATEMENTS ..........
      DOUBLE PRECISION  START,     STOP,      TOL
      INTEGER           FMT,       FRSTR,     FRSTP,     ANSTAT
      INTEGER PRNDX1,PRNDX2,PRNDX3
      INTEGER       ANCNTR,              ANREC
C
C     .......... DIMENSION STATEMENTS ..........
C
      DIMENSION IIN(102),RIN(28)
C
C     .......... COMMON STATEMENTS ..........
C
      COMMON / CALRT /  JJNT,             NDICAL,            NTRCAL
      COMMON/PRNDX/PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *              ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *              IWLFG, IRAMFG,LZNE,IPG
      COMMON /DCARGN/ START,  STOP,      TOL,       IFLAGG,
     1                FMT,    IDKRTP,    FRSTR(2),  FRSTP(2),
     2                IDBLE,  LU,        ISIZE,     NTH,
     3                MAX,    NAV,       INIT,      IDDC,
     4                ISTAT
      COMMON /FLDAT/  J,K,LF,NM
      COMMON /INPUT/ IIN(102),RIN(28)
      COMMON / DCDATA /  MEAS(3993)
C
      DATA LMDI/80/,LMTR/60/
C
C
      IF(JJNT .EQ. 0)                                  GOTO 1010
      IMAX6=IIN(73)
      IMIN6=IIN(74)
      IIN(73)=NDICAL+LMDI
      IIN(74)=NDICAL-LMDI
      IMAX12=IIN(85)
      IMIN12=IIN(86)
      IIN(85)=NTRCAL+LMTR
      IIN(86)=NTRCAL-LMTR
C
C
 1010 IF(PRNDX1(1,1) .EQ. 0)GOTO 1020
      NMEAS=82
      GOTO 1030
 1020 NMEAS=78
 1030 LF=231
      DO 1200 J=63,NMEAS,2
      DO 1100 K=1,MAX
C
```

```
C
      IF(J .LE. 78)                                         GOTO 1040
      IF(K .LT. PRNDX1(1,1))GOTO 1100
      IF(K .GT. PRNDX1(2,1))                                GOTO 1150
C
C
 1040 IF(MEAS(LF+K) .GT. IIN(J))                            GOTO 1250
      IF(MEAS(LF+K) .LT. IIN(J+1))                          GOTO 1250
      KF=0
C
 1100 CONTINUE
 1150 LF=LF+MAX
 1200 CONTINUE
      GOTO 1300
C
C
 1250 LRET=1
      JPAR=(J-61)/2
      GOTO 1900
 1260 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 1100
C
C
 1300 LF=MEAS(179)-1
      DO 1400 K=1,MAX
      IF(K .LT. PRNDX1(1,1))GOTO 1330
      IF(K .LT. PRNDX1(1,4))GOTO 1310
      IF(K .GT. PRNDX2(2,4))GOTO 1310
      NM=91
      GOTO 1390
C
C
C
 1310 IF(K .LT. PRNDX1(1,5))GOTO 1320
      IF(K .GT. PRNDX1(2,5))GOTO 1320
      NM=93
      GOTO 1390
C
C
 1320 IF(K .LT. PRNDX1(1,6))GOTO 1330
      IF(K .GT. PRNDX1(2,6))GOTO 1330
      NM=95
      GOTO 1390
C
C
 1330 NM=89
 1390 IF(MEAS(LF+K) .GT. IIN(NM))                           GOTO 1410
      IF(MEAS(LF+K) .LT. IIN(NM+1))                         GOTO 1410
      KF=0
 1400 CONTINUE
C
      GOTO 1430
C
 1410 LRET=2
```

```
      JPAR=11
      GOTO 1900
 1420 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 1400
C
C
 1430 LF=MEAS(180)-1
      DO 1490 K=1,MAX
      DO 1470 J=97,102,2
C
C
      IF(MEAS(LF+K) .GT. IIN(J))                                  GOTO 1470
      IF(MEAS(LF+K) .LT. IIN(J+1))                                GOTO 1470
      KF=0
      GOTO 1490
 1470 CONTINUE
      GOTO 1500
 1490 CONTINUE
      GOTO 1520
 1500 LRET=3
      JPAR=12
      GOTO 1900
 1510 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 1490
C
C
 1520 LF=MEAS(181)-1
      NM=MEAS(182)-1
      J=MAX*2
      DO 1600 K=1,J
      IF(K .LT. PRNDX2(1,3))GOTO 1530
      IF(K .GT. PRNDX2(2,3))GOTO 1530
      IHIGH=1023
      ILOW=998
      GOTO 1540
 1530 IHIGH=IIN(83)
      ILOW=IIN(84)
 1540 IF(MEAS(LF+K) .GT. IHIGH)GOTO 1550
      IF(MEAS(LF+K) .LT. ILOW)GOTO 1550
      KF=0
      GOTO 1570
C
C
C
 1550 LRET=4
      JPAR=13
      GOTO 1900
 1560 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
 1570 IF(MEAS(NM+K) .GT. IIN(85))                                 GOTO 1580
      IF(MEAS(NM+K) .LT. IIN(86))                                 GOTO 1580
```

```
      KF=0
      GOTO 1600
C
C
 1580 LRET=5
      JPAR=14
      GOTO 1900
 1590 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
 1600 CONTINUE
 1610 IF(PRNDX1(1,5) .LE. 0)GOTO 1980
      LF=MEAS(189)-1
C
C
      J=PRNDX1(1,5)
      NM=PRNDX1(2,5)
      DO 1700 K=J,NM
C
      IF(MEAS(LF+K) .GT. IIN(87))                              GOTO 1650
      IF(MEAS(LF+K) .LT. IIN(88))                              GOTO 1650
      KF=0
      GOTO 1700
C
 1650 LRET=6
      JPAR=15
      GOTO 1900
 1660 NSAM=K
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
 1700 CONTINUE
      GOTO 1980
C
C
C
C
 1900 IF(MACT .LE. 0)GOTO 1950
      IF(ANSTAT(JPAR) .EQ. 1)GOTO 1960
      IF(K .EQ. 1)                                             GOTO 1910
      IF(KF .EQ. 0)GOTO 1910
      GOTO 1920
 1910 KF=K
      GOTO 1950
 1920 IF((KF+1) .EQ. K)GOTO 1960
      KF=0
 1950 LAN=0
      GOTO 1970
 1960 ANSTAT(JPAR)=1
      LAN=1
 1970 GOTO(1260,1420,1510,1560,1590,1660)LRET
C
C
 1980 IF(MACT .LE. 0)GOTO 1999
      JPAR=0
      NSAM=0
      LAN=0
```

```
      KRET=0
      CALL SEG('DK1:ANPRO.LLL',2)
 1999 CONTINUE
      CALL SEG('DK1:QAEXEC.GGG',0,0)
      END

ROUTINES CALLED:
SEG

SWITCHES = /ON,/SU,/CO

BLOCK        LENGTH
MAIN.   1198    (004534)*
CALRT   3       (000006)
PRNDX   54      (000154)
ANDAT   91      (000266)
DCARGN  28      (000070)
FLDAT   4       (000010)
INPUT   158     (000474)
DCDATA  3993    (017462)

**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES  04331 12935
EXECUTABLES   04711 12555
ASSEMBLY      01942 18241
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:QA191H.OBJ,LP:<DK1:QA191H.FTN/ON/SU/CO:99
```

```
C       ****************************************************
C       *   RESIDENT PROGRAM FOR S191H DATA QUALITY TEST   *
C       ****************************************************
C
C
        DOUBLE PRECISION   TMCURR,TMFRST,TMFRCR
        DOUBLE PRECISION  STTME
        DOUBLE PRECISION LSTTME
        DOUBLE PRECISION SSTIME,START,STOP,TOL
        DOUBLE PRECISION SUMTM
C
C
        INTEGER ANSTAT,  ANCNTR,  ANREC
        INTEGER PRNDX1,PRNDX2,PRNDX3,FMT,FRSTR,FRSTP
C
C
C
        COMMON /A4CHNL/ IA4(685)
        COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
       *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
       *               IWLFG, IRAMFG,LZNE,IPG
        COMMON / CALRT /  JJNT,                NDICAL,            NTRCAL
        COMMON /CMPLET/ NCMPLT(2)
        COMMON /WVLDAT/  MCMPLT(2),  LMTLO,  LMTHI
        COMMON /LASFRM/ LSTTME
        COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
       *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
        COMMON /BIADAT/ IBHIS1(7),  IBHIS2(7),  IBHIS3(7),  IBHIS5(7),
       *                IBHIS6(7),  ISSN,  LCMPLT(2)
        COMMON /DCARGN/ START,  STOP,     TOL,      IFLAGG,
       1                FMT,    IDKRTP,   FRSTR(2), FRSTP(2),
       2                IDBLE,  LU,       ISIZE,    NTH,
       3                MAX,    NAV,      INIT,     IDDC,
       4                ISTAT
        COMMON /DCDATA /MEAS(3993)
        COMMON /ERROR/  IERR(20)
        COMMON /FLDAT/  JZ,KZ,LZF,NZM
        COMMON /INPUT/  IIN(102),          RIN(28)
        COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
        COMMON /PRNDX/  PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
        COMMON /QADAT/ STTME,    MFLG
        COMMON /RECPTR/ IVAR2
        COMMON /RESDAT/ JCMPLT(2),KCMPLT(2),NA4(685),NFSCN,NX,NSAMPL,NIX
       *, AVJ2, AVJ3, AVJ5, AVJ6, SVJ2, SVJ3, SVJ5, SVJ6, JSCN6, NSSCAN
       *,   IPSOK
        COMMON / SAVE   /  IBGNSC        ,NMVALD        ,IAHEAD        ,
       1                   IPRVOK        ,ISAVPT        ,IA4NDX        ,
       2                   TMCURR        ,TMFRCR        ,TMFRST        ,
       3                   SUMTM         ,RMSTOT        ,IA6SET        ,
       4                   RMNSUM        ,RMXSUM        ,RXSMSQ        ,
       5                   RNSMSQ
        COMMON  /SIXSV/  ISIX
        COMMON /TIMES/  BJD,    SSTIME(2,65)
        COMMON /TITLES/ ITITL(32)
C
```

```
C
      CALL NEWMMC('AAA')
      CALL SEG('DK1:CONDRV.FFF',0,0)
      IF(IERR(1) .EQ.69)                                       GOTO 9600
C
      CALL NEWMMC('BBB')
      CALL SEG('DK1:QAEXEC.GGG',1,1)
C
 9600 CALL SEGEND
      CALL EXIT
      STOP
      END

ROUTINES CALLED:
NEWMMC, SEG   , SEGEND, EXIT

SWITCHES = /ON,/SU,/CO

BLOCK       LENGTH
MAIN.   180    (000550)*
A4CHNL  685    (002532)
ANDAT   91     (000266)
CALRT   3      (000006)
CMPLET  2      (000004)
WVLDAT  4      (000010)
LASFRM  4      (000010)
HISDAT  8      (000020)
BIADAT  38     (000114)
DCARGN  28     (000070)
DCDATA  3993   (017462)
ERROR   20     (000050)
FLDAT   4      (000010)
INPUT   158    (000474)
INTNDX  5      (000012)
PRNDX   54     (000154)
QADAT   5      (000012)
RECPTR  1      (000002)
RESDAT  712    (002620)
SAVE    33     (000102)
SIXSV   1      (000002)
TIMES   522    (002024)
TITLES  32     (000100)

**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES  03750 13516
EXECUTABLES   05159 12107
ASSEMBLY      01581 18602
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:QAEXEC.OBJ,LP:<DK1:QAEXEC.FTN/ON/SU/CO:99
```

```
C**********************************************************************
C*                                                                    *
C*        QAEXEC                                                      *
C*                                                                    *
C**********************************************************************
C
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *               IWLFG, IRAMFG,LZNE,IPG
C
      COMMON / CALRT /  JINT,               NDICAL,             NTRCAL
C
      COMMON / CMPLET / NCMPLT(2)
C
      COMMON /DCARGN/ START,  STOP,     TOL,      IFLAGG,
     1                FMT,    IDKRTP,   FRSTR(2), FRSTP(2),
     2                IDBLE,  LU,       ISIZE,    NTH,
     3                MAX,    NAV,      INIT,     IDDC,
     4                ISTAT
C
      COMMON / DCDAT/   MEAS(3993)
C
      COMMON / ERROR /  IERR(20)
C
      COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
     *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
C
      COMMON / INPUT /  IIN(102),           RIN(28)
C
      COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
      COMMON /LASFRM/ LSTTME
C
      COMMON / PRNDX /  PRNDX1(2,9),        PRNDX2(2,9),
     1                  PRNDX3(2,9)
C
      COMMON /RECPTR/ IVAR2
      COMMON /QADAT/ STTME,    MFLG
C
      COMMON / TIMES /  BJD,                SSTIME(2,65)
C
      COMMON / TITLES / ITITL(32)
C
      DIMENSION ITIM(2)
C
      EQUIVALENCE       (IIN(11),IHTAB)
      EQUIVALENCE       (IIN(12),IDELOP)
      EQUIVALENCE       (IIN(13),IDELHR)
      EQUIVALENCE       (IIN(14),IDELMN)
      EQUIVALENCE       (IIN(15),IDELSC)
      EQUIVALENCE       (IIN(16),NOVRL)
      DOUBLE PRECISION  SSTIME,             START
      DOUBLE PRECISION LSTTME
      DOUBLE PRECISION  STOP,               TOL
      DOUBLE PRECISION  LASPRC, FRSPRC, DELTME,TMS153,DELT
      DOUBLE PRECISION  STTME
```

```
C
      INTEGER          PRNDX1,          PRNDX2,          PRNDX3
      INTEGER          FMT,             FRSTR,           FRSTP
      INTEGER          ANSTAT
      INTEGER ANCNTR,ANREC
      BYTE BDATE(9)
      DIMENSION        NTABUF(50)
C
      DATA    U/'U'/
C
      CALL SETEMT
      CALL SETFIL(2,'ARR.TMP',IERR,'SY',0,0,0,2)
      DEFINE FILE 2(5,685,U,IVAR2)
      CALL NTRAN(12,5)
      MFLG=0
      IDDC=01
      IF(IDELOP .EQ. 1 .OR. IHTAB .EQ. 2)                GO TO 5900
      IFLAGG=0
      IDKRTP=1
      FRSTR(1)=0
      IDBLE=1
      LU=9
      ISIZE=3993
      NTH=1
      MAX=18
      NAV=1
      INIT=0
      IERR(1)=0
C
C
      CALL SEG('DK1:DECRIP,RRR',2)
      IF(ISTAT .EQ. 0)                                   GO TO 5010
      WRITE(6,9000)ISTAT
      CALL SEGRET
C
 5010 IFLAGG=0
      FMT = 27
      TOL=30.0D0
      KO=1
 5030 IPIT=0
      START=SSTIME(1,KO)
      IF(IWLHDR .EQ. 1 .AND. IWLFLG .EQ. 0)              GOTO 5040
      IWLHDR=0
      IWLFLG=0
 5040 STOP=SSTIME(2,KO)
      IF(IBVHDR .NE. 1)                                  GOTO 5050
      IF(IBVFLG .NE. 1)                                  GOTO 5050
      IF(PRNDX3(1,3) .GT. 0)                             GOTO 5050
      IBVHDR=0
      IBVFLG=0
 5050 IF(ICLHDR .NE. 1)                                  GOTO 5070
      IF(IDONE .NE. 12)                                  GOTO 5070
      IF(PRNDX3(1,1) .GT. 0)                             GOTO 5070
      ICLHDR=0
      IDONE=0
      IRSFLG=0
```

```
      IRMFLG=0
C
 5070 MAX=18
      CALL SEG('DK1:DCMDRV.III',2)
      IF (IIN(3) .GE. 0)                                 GOTO 5071
      CALL PDUMP(START,START,5)
      CALL PDUMP(STOP,STOP,5)
      CALL PDUMP(ISTAT,ISTAT,1)
      CALL PDUMP(MAX,MAX,1)
 5071 CONTINUE
      IF(MFLG .EQ. 1)                                    GOTO 5075
      CALL AMOV(MEAS(69),STTME,4)
      CALL AMOV(MEAS(85),FRSPRC,4)
      RNN=MAX/6
      ISE=RNN
      STTME=STTME+(FRSPRC*ISE)
      MFLG=1
 5075 IF(MAX .EQ. 0)                                     GOTO 5750
C
      IF(ISTAT .EQ. 0)                                   GO TO 5130
      IF(ISTAT .EQ. 1)                                   GO TO 5100
      IF(ISTAT .EQ. 2)                                   GO TO 5120
      IF(ISTAT .EQ. 6)                                   GO TO 5030
      IF(ISTAT .EQ. 7)                                   GO TO 5030
      IF(ISTAT .EQ. 8)                                   GO TO 5030
      CALL SEG('DK1:ERRDRV.HHH',2)
                                                         GO TO 9999
 5100 IF(MACT .EQ. 0)                                    GO TO 5110
      JPAR=0
      LAN=FRSTR(1)-2
      KPET=4
      N3AM=MAX
      CALL SEG('DK1:ANPRO.LLL',2)
 5110 IF(STOP .LT. SSTIME(2,KO))                         GO TO 5130
 5120 KO=KO+1
      IF(KO .GT. NOVRL)                                  GO TO 5760
      IPIT=1
C
 5130 CALL SEG('DK1:TMLOOP.PPP',2)
C
      CALL SEG('DK1:FLDPRO.JJJ',2)
C
      IF(PRNDX3(1,9) .EQ. 0)                             GO TO 5870
      IF(IWLHDR .EQ.1 .OR. IWLFLG .EQ. 1)                GOTO 5867
 5135 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=19
      NTABUF(2)=IIN(10)
      NTABUF(3)=IIN(9)
      NTABUF(4)=IIN(8)
      TMS153=SSTIME(1,LWLW)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(8)=ITIM(1)
      NTABUF(9)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
```

```
      NTABUF(10)=INTNUM
      NTABUF(11)=IFRAC
      TMS153=SSTIME(2,LWLW)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5136 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5137,5136,5139
 5137 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                          GOTO 9999
 5139 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=20
      CALL AMOV(MEAS(69),TMS153,4)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5860 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5861,5860,5862
 5861 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                          GOTO 9999
 5862 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=21
      NTABUF(2)=IABS(IIN(3))
      CALL AMOV(IIN(4),NTABUF(3),4)
      CALL NTRAN(12,1,50,NTABUF(1))
 5863 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5864,5863,5865
 5864 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                          GOTO 9999
 5865 IWLHDR=1
 5867 CALL SEG('DK1:WVLPRO.MMM',2)
                                                          GOTO 5140
 5870 IF(IWLHDR .EQ. 1 .AND. IWLFLG .EQ. 0)               GOTO 5867
C
 5140 IF(PRNDX3(1,1) .EQ. 0)                              GO TO 5850
 5245 IF(ICLHDR .EQ. 1)                          GOTO 5807
C     ***********************************************************
C     *   WRITE CAL PERIOD HEADER TO SCRATCH TAPE                 *
C     ***********************************************************
      CALL ZERO(NTABUF(1),50)
      NTABUF(1)=1
```

```
      NTABUF(2)=IIN(10)
      NTABUF(3)=IIN(9)
      NTABUF(4)=IIN(8)
      TMS153=SSTIME(1,L)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(8)=ITIM(1)
      NTABUF(9)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(10)=INTNUM
      NTABUF(11)=IFRAC
      TMS153=SSTIME(2,L)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5146 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5147,5146,5149
 5147 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                  GOTO 9999
 5149 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=2
      CALL AMOV(MEAS(69),TMS153,4)
      CALL AMOV(MEAS(85),DELT,4)
 5777 IF(TMS153 .GE. SSTIME(1,L))                     GOTO 5888
      TMS153=TMS153+DELT
                                                  GOTO 5777
 5888 CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5800 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5801,5800,5802
 5801 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                  GOTO 9999
 5802 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=3
      NTABUF(2)=IABS(IIN(3))
      CALL AMOV(IIN(4),NTABUF(3),4)
      CALL NTRAN(12,1,50,NTABUF(1))
 5751 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5752,5751,5806
 5752 LULU=12
```

```
      WRITE(6,9050)LULU,LSTAT
                                                          GOTO 9999
 5806 IDONE=3
      ICLHDR=1
 5807 CALL SEG('DK1:RAMPRO,NNN',2)
                                                          GOTO 5150
 5850 IF(ICLHDR .EQ. 0)                                   GOTO 5150
      IF(IRMFLG .EQ. 1)                                   GOTO 5851
      CALL SEG('DK1:RAMPRO,NNN',2)
 5851 IF(IRSFLG .EQ. 2)                                   GOTO 5150
      CALL SEG('DK1:RESPRO,000',2)
C
 5150 IF(PRNDX3(1,7) .EQ. 0 .AND. PRNDX3(1,8) .EQ. 0) GO TO 5200
 5155 CALL SEG('DK1:RESPRO,000',2)
C
 5200 IF(PRNDX3(1,3) .EQ. 0)                              GO TO 5300
      IF(IBVHDR .GT. 0 .OR. IBVFLG .GT. 0)                GOTO 5250
      CALL ZERO(NTABUF(1),50)
      NTABUF(1)=23
      NTABUF(2)=IIN(10)
      NTABUF(3)=IIN(9)
      NTABUF(4)=IIN(8)
      TMS153=SSTIME(1,LBIAS)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(8)=ITIM(1)
      NTABUF(9)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(10)=INTNUM
      NTABUF(11)=IFRAC
      TMS153=SSTIME(2,LBIAS)
      CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5201 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5202,5201,5203
 5202 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                          GOTO 9999
 5203 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=2
      CALL AMOV(MEAS(69),TMS153,4)
      CALL AMOV(MEAS(85),DELT,4)
 5204 IF(TMS153 .GE. SSTIME(1,LBIAS))                     GOTO 5205
      TMS153=TMS153+DELT
                                                          GOTO 5204
 5205 CALL STOHMS(TMS153,ITIM,SEC)
      NTABUF(12)=ITIM(1)
      NTABUF(13)=ITIM(2)
```

```
      INTNUM=SEC
      RFRAC=SEC-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 5206 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5207,5206,5208
 5207 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                    GOTO 9999
 5208 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=3
      NTABUF(2)=IABS(IIN(3))
      CALL AMOV(IIN(4),NTABUF(3),4)
      CALL NTRAN(12,1,50,NTABUF(1))
 5209 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5210,5209,5211
 5210 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                    GOTO 9999
 5211 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=13
      CALL AMOV(IIN(26),NTABUF(2),35)
      CALL NTRAN(12,1,50,NTABUF(1))
 5212 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)5213,5212,5214
 5213 LULU=12
      WRITE(6,9050)LULU,LSTAT
                                                    GOTO 9999
 5214 IBVHDR=1
 5250 CALL SEG('DK1:BIAPRO.KKK',2)
C
 5300 IF(IPIT .GT. 0)                              GO TO 5030
      CALL AMOV(MEAS(153),TMS153,4)
      START = STOP + TMS153
                                                   GO TO 5040
 5750 KO = KO + 1
      IF(KO .LE. NOVRL)                            GO TO 5030
 5760 IF (IIN(17) .EQ. 0 .AND. IIN(18) .EQ. 0)     GO TO 9999
 5900 CALL SEG('DK1:QASUM.QQQ',2)
C
C
C
C     *****************************************************************
C     *    FORMAT STATEMENTS                                          *
C     *****************************************************************
 9000 FORMAT(1H ,'* * * * * ERROR - WHILE TRYING TO READ TAPE DESCRIPTOR
     1 FILE A STATUS OF',I5,' WAS RETURNED')
 9050 FORMAT              (1H0,'NTRAN WRITE ERROR ON UNIT',I4,
     *                         'STATUS WORD =',I4)
 9150 FORMAT              (1H0,'NTRAN READ ERROR ON UNIT',I4,
     *                         'STATUS WORD =',I4)
C     *****************************************************************
C
C
C
```

```
FORTRAN V004A                     17:34:06    27-JUL-76    PAGE    8

        9999 CALL SEGRET
             END

       ROUTINES CALLED:
       SETEMT, SETFIL, NTRAN , SEG   , SEGRET, PDUMP , AMOV
       ZERO  , STOHMS, IABS

       SWITCHES = /ON,/SU,/CO

       BLOCK       LENGTH
       MAIN.   2901   (013252)*
       ANDAT   91     (000266)
       CALRT   3      (000006)
       CMPLET  2      (000004)
       DCARGN  28     (000070)
       DCDATA  3993   (017462)
       ERROR   20     (000050)
       HISDAT  8      (000020)
       INPUT   158    (000474)
       INTNDX  5      (000012)
       LASFRM  4      (000010)
       PRNDX   54     (000154)
       RECPTR  1      (000002)
       QADAT   5      (000012)
       TIMES   522    (002024)
       TITLES  32     (000100)

       **COMPILER ----- CORE**
          PHASE       USED  FREE
       DECLARATIVES 03750 13516
       EXECUTABLES  05433 11833
       ASSEMBLY     02726 17457
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1: QASUM.OBJ,LP:<DK1: QASUM.FTN/ON/SU/CO:99
```

```
C***********************************************************************
C*                                                                     *
C*       QASUM                                                         *
C*                                                                     *
C***********************************************************************
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *               IWLFG, IRAMFG,LZNE,IPG
C
      COMMON / CALRT /  JINT,                NDICAL,              NTRCAL
C
      COMMON / CMPLET / NCMPLT(2)
C
      COMMON /DCARGN/ START,  STOP,      TOL,       IFLAGG,
     1                FMT,    IDKRTP,    FRSTR(2),  FRSTP(2),
     2                IDBLE,  LU,        ISIZE,     NTH,
     3                MAX,    NAV,       INIT,      IDDC,
     4                ISTAT
C
      COMMON / DCDATA /  MEAS(3993)
C
      COMMON / ERROR /  IERR(20)
C
      COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
     *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
C
      COMMON / INPUT /  IIN(102),            RIN(28)
C
      COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
C
      COMMON / PRNDX /  PRNDX1(2,9),         PRNDX2(2,9),
     1                  PRNDX3(2,9)
C
      COMMON /QADAT/ STTME,    MFLG
C
      COMMON / TIMES /  BJD,                 SSTIME(2,65)
C
      COMMON / TITLES / ITITL(32)
C
C
      EQUIVALENCE       (IIN(11),IHTAB)
      EQUIVALENCE       (IIN(12),IDELOP)
      EQUIVALENCE       (IIN(13),IDELHR)
      EQUIVALENCE       (IIN(14),IDELMN)
      EQUIVALENCE       (IIN(15),IDELSC)
C
      DOUBLE PRECISION  SSTIME,              START
      DOUBLE PRECISION  STOP,                TOL
      DOUBLE PRECISION  LASPRC, FRSPRC, DELTME,TMS153,DELT
      DOUBLE PRECISION  STTME
C
      INTEGER           PRNDX1,              PRNDX2,              PRNDX3
      INTEGER           FMT,                 FRSTR,               FRSTP
      INTEGER           ANSTAT
      INTEGER ANCNTR,ANREC
```

```
        BYTE BDATE(9)
C
        DIMENSION ITIM(2)
        DIMENSION          NTABUF(50)
        DIMENSION          W(19)
        DIMENSION          NTBSTO(900)
        DIMENSION          NTEMP(50)
        DIMENSION RMON(12),ICHNLA(6)
        DATA ICHNLA/ 'A1',     'A2',     'A3',' ','A5',     'A6' /
        DATA RMON/
       *    'JAN-',   'FEB-',   'MAR-',   'APR-',   'MAY-',   'JUN-',
       *    'JUL-',   'AUG-',   'SEP-',   'OCT-',   'NOV-',   'DEC-'/
        IPAGE = 0
        CALL DATE(BDATE)
        IF (IDELOP .EQ. 1)                                  GO TO 5950
        IF (IHTAB .EQ. 2)                                   GO TO 8000
        IPAGE=1
        CALL AMOV(MEAS(69),FRSPRC,4)
        CALL STOHMS(STTME,ITIM,SEC)
        IFFH=ITIM(1)
        IFFM=ITIM(2)
        RFFS=SEC
        I3ABS=IABS(IIN(3))
        CALL STOHMS(FRSPRC,ITIM,SEC)
        WRITE(6,9400)IPAGE,BDATE,IFFH,IFFM,RFFS,
       *              ITIM(1),ITIM(2),SEC,
       *               I3ABS,IIN(4),IIN(5),IIN(6),IIN(7),
       *              TANCTR(17),TANCTR(18),
       *              TANCTR(16),TANCTR(15),
       *              TANCTR(1),TANCTR(2),
       *              TANCTR(3),TANCTR(4),
       *              TANCTR(13),TANCTR(5),
       *              TANCTR(6),TANCTR(9),
       *              TANCTR(14),TANCTR(10),
       *              TANCTR(7),TANCTR(8),
       *              TANCTR(11),TANCTR(12)
        IF(IWLFG .EQ. 0)                                    GOTO 5910
        WRITE(6,9450)
   5910 IF(IRESFG .EQ. 0)                                   GOTO 5915
        WRITE(6,9460)
   5915 IF(IRAMFG .EQ. 0)                                   GOTO 5920
        WRITE(6,9465)
                                                            GOTO 5930
   5920 WRITE(6,9470)
   5930 CALL ZERO(TANCTR(1),36)
        IWLFG=0
        IRESFG=0
        STTME=0.
        FRSPRC=0.
C     ******************************************************************
C     *    TAB OUT REQUIRED FILES AND BUILD NEW HISTORICAL FILE TAPE   *
C     ******************************************************************
C
C
C     ******************************************************************
C     *    INITIALIZATION PHASE:CLEAR COUNTERS AND FLAGS; REWIND TAPES  *
```

```
C     ******************************************************************
 5950 JKTAB=0
      NDOLD = 0
      INSERT= 0
      CALL ZERO(NTEMP(1),50)
      CALL ZERO(NTBSTO,900)
      IF(IDELOP .EQ. 1)                                  GO TO 5975
      CALL NTRAN(9,5)
      CALL NTRAN(12,4)
      CALL NTRAN(12,4)
      CALL NTRAN(12,4)
      CALL NTRAN(12,5)
 5975 CALL NTRAN(10,5)
      CALL NTRAN(11,5)
 6000 LIN = 10
      LOUT= 11
      LSCR= 12
C
C
C     ******************************************************************
C     *     READ OLD HISTORICAL FILE TAPE                              *
C     ******************************************************************
 6010 CALL NTRAN(LIN,2,50,NTABUF)
C
C
C     ******************************************************************
C     *     CHECK STATUS OF THE UNIT                                   *
C     *       0   = OPERATION COMPLETE                                 *
C     *      -1   = LAST TRANSMISSION NOT COMPLETED                    *
C     *      -2   = END OF FILE DETECTED                               *
C     *      -3   = DEVICE ERROR                                       *
C     *      -4   = TRANSMISSION ABORTED                               *
C     *      -5   = END OF MEDIUM DETECTED                             *
C     *     IF PREVIOUS OPERATION WAS A READ OR WRITE,AND NO ERROR     *
C     *     OCCURRED, LSTAT WILL CONTAIN THE NUMBER OF 16-BIT WORDS    *
C     *     TRANSFERRED.                                               *
C     ******************************************************************
 6020 CALL NTRAN(LIN,15,LSTAT)
      IF (LSTAT + 1) 6030,6020,6060
 6030 IF (LSTAT .EQ. -2)                                 GO TO 6050
      WRITE(6,9150) LIN,LSTAT
                                                         GO TO 9999
 6050 NDOLD = 1
      IF (INSERT .NE. 1 .AND. IDELOP .NE. 1)             GO TO 7000
 6055 CALL NTRAN(LOUT,4)
      CALL NTRAN(LOUT,4)
      CALL NTRAN(LOUT,4)
      CALL NTRAN(LIN,5)
      CALL NTRAN(LOUT,5)
      IF(IDELOP .EQ. 1)                                  GO TO 6059
      CALL NTRAN(LSCR,5)
 6059 WRITE (6,9500)
                                                         GO TO 9999
 6060 IF (IIN(3) .GE. 0)
     *                                                   GO TO 6065
      CALL PDUMP(NTABUF(1),NTABUF(41),1)
```

```
 6065 IF (NTABUF(1) .NE. 1 .AND. NTABUF(1) .NE. 19 .AND.
     *    NTABUF(1) .NE. 23)                               GO TO 6700
C     ****************************************************************
C     *    DATE RECORD TYPE HAS BEEN READ                            *
C     ****************************************************************
C     ****************************************************************
C     *    FILE DELETION CHECK                                       *
C     ****************************************************************
 6070 IF (IDELOP .EQ. 0)                                   GO TO 6100
      IF(IIN(8) .NE. NTABUF(4) .OR. IIN(9) .NE. NTABUF(3)
     * .OR. IIN(10) .NE. NTABUF(2))                        GO TO 6705
      IF (NTABUF(8) .NE. IDELHR .OR. NTABUF(9) .NE. IDELMN .OR.
     *    NTABUF(10) .NE. IDELSC)                          GO TO 6705
C     ****************************************************************
C     *    DELETE THE FILE                                           *
C     ****************************************************************
 6075 CALL NTRAN(LIN,2,50,NTABUF)
 6077 CALL NTRAN(LIN,15,LSTAT)
      IF (LSTAT + 1)6080,6077,6085
 6080 IF (LSTAT .EQ. -2)                                   GO TO 6050
      WRITE (6,9150) LIN,LSTAT
                                                           GO TO 9999
 6085 IF (IIN(3) .GE. 0)
     *                                                     GO TO 6090
      CALL PDUMP(NTABUF(1),NTABUF(41),1)
 6090 IF (NTABUF(1) .NE. 1 .AND. NTABUF(1) .NE. 19 .AND.
     *    NTABUF(1) .NE. 23)                               GO TO 6075
      JKTAB = 0
                                                           GO TO 6070
C     ****************************************************************
C     ****************************************************************
C
C
C
C
C
 6100 IF (IIN(8) .LT. NTABUF(4))                           GO TO 6600
      IF (IIN(8) .EQ. NTABUF(4) .AND. IIN(9)
     *                         .LT. NTABUF(3))             GO TO 6600
      IF (IIN(9) .EQ. NTABUF(3) .AND. IIN(10)
     *                         .LT. NTABUF(2))             GO TO 6600
      IF (IIN(3) .GE. 0)
     *                                                     GO TO 6700
      CALL PDUMP(NTABUF(2),NTABUF(4),1)
                                                           GO TO 6700
 6600 IF (INSERT .NE. 0)                                   GO TO 6705
      CALL AMOV(NTABUF(1),NTEMP(1),50)
                                                           GO TO 7000
C     ****************************************************************
C     *    WRITE A RECORD TO THE NEW HISTORICAL TAPE                 *
C     ****************************************************************
 6700 IF (NTEMP(1) .EQ. 0)                                 GO TO 6705
      CALL AMOV(NTEMP(1),NTABUF(1),50)
      CALL ZERO(NTEMP,50)
 6705 CALL NTRAN(LOUT,1,50,NTABUF(1))
C     ****************************************************************
C     *    PERFORM STATUS CHECKS                                     *
```

```
C     ******************************************************************
 6710 CALL NTRAN(LOUT,15,LSTAT)
      IF(LSTAT + 1)6720,6710,6730
 6720 WRITE (6,9050) LOUT,LSTAT
                                                          GO TO 9999
C     ******************************************************************
C     *    TEST TO SEE IF TAB IS REQUIRED                              *
C     ******************************************************************
 6730 IF (IHTAB .NE. 1)                                   GO TO 6010
 6740 IF (NTABUF(1) .NE. 23)                              GO TO 6745
      KMTAB = 1
                                                          GO TO 6780
 6745 IF (NTABUF(1) .GT. 18)                              GO TO 6750
      KMTAB = ((NTABUF(1) - 1) * 50) + 1
                                                          GO TO 6780
C     ******************************************************************
C     *    TRANSFER THE DATA FROM NTABUF TO NTBSTO                     *
C     ******************************************************************
 6750 KMTAB = ((NTABUF(1) - 19)* 50) + 1
 6780 CALL AMOV(NTABUF(1),NTBSTO(KMTAB),50)
      IF (IIN(3) .GE. 0)
     *                                                    GO TO 6790
      CALL PDUMP(NTABUF(1),NTABUF(41),1)
C     ******************************************************************
C     *    UPDATE THE RECORD COUNTER FOR THE TAB                       *
C     ******************************************************************
 6790 JKTAB = JKTAB + 50
C     ******************************************************************
C     *    DOES NTBSTO CONTAIN A COMPLETE FILE ?                       *
C     ******************************************************************
      IF (NTBSTO(1) .EQ. 1 .AND. JKTAB .EQ. 600)          GO TO 6830
      IF (NTBSTO(1) .EQ.19 .AND. JKTAB .EQ. 200)          GO TO 6850
      IF (NTBSTO(1) .EQ.23 .AND. JKTAB .EQ. 450)          GO TO 6830
                                                          GO TO 6010
C
C
C     ******************************************************************
C     *    CALIBRATION PERIOD IS BEING PROCESSED                       *
C     ******************************************************************
 6830 IPAGE = IPAGE + 1
      KMLO = 595
      KMHI = 601
      KCHN = 602
      L1 = 607
      MO = NTBSTO(3)
      WRITE (6,9200)      IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
     *                    (NTBSTO(MN),
     *                    MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
     *                    MK=62,65)
      IF (NTBSTO(1) .EQ. 23)                              GO TO 6834
      WRITE (6,9210)
      WRITE (6,9230)
      W( 1) = (NTBSTO(165) * .0001) + FLOAT(NTBSTO(164))
      W( 2) = (NTBSTO(220) * .0001) + FLOAT(NTBSTO(219))
      W( 3) = (NTBSTO(265) * .0001) + FLOAT(NTBSTO(264))
      W( 4) = (NTBSTO(205) * .0001) + FLOAT(NTBSTO(204))
```

```
      W( 5) = (NTBSTO(365) * .0001) + FLOAT(NTBSTO(364))
      W( 6) = (NTBSTO(210) * .0001) + FLOAT(NTBSTO(209))
      W( 7) = (NTBSTO(465) * .0001) + FLOAT(NTBSTO(464))
      W( 8) = (NTBSTO(215) * .0001) + FLOAT(NTBSTO(214))
      W( 9) = (NTBSTO(565) * .0001) + FLOAT(NTBSTO(564))
      W(10) = (NTBSTO(305) * .0001) + FLOAT(NTBSTO(304))
      W(11) = (NTBSTO(567) * .0001) + FLOAT(NTBSTO(566))
      W(12) = (NTBSTO(310) * .0001) + FLOAT(NTBSTO(309))
      W(13) = (NTBSTO(315) * .0001) + FLOAT(NTBSTO(314))
      W(14) = (NTBSTO(405) * .0001) + FLOAT(NTBSTO(404))
      W(15) = (NTBSTO(410) * .0001) + FLOAT(NTBSTO(409))
      W(16) = (NTBSTO(415) * .0001) + FLOAT(NTBSTO(414))
      W(17) = (NTBSTO(505) * .0001) + FLOAT(NTBSTO(504))
      W(18) = (NTBSTO(510) * .0001) + FLOAT(NTBSTO(509))
      W(19) = (NTBSTO(515) * .0001) + FLOAT(NTBSTO(514))
 6833 WRITE (6,9240) (W(NK),NK=1,19)
                                                          GO TO 6900

 6834 WRITE (6,9220)
      DO 6835 IJ=1,5
      KMLO = KMLO + 7
      KMHI = KMHI + 7
      KCHN = KCHN + 50
      L1 = L1 + 50
      L7 = L1 +  6
      ICHN = NTBSTO(KCHN)
      WRITE (6,9245)
      WRITE (6,9250)    (NTBSTO(KM),KM=KMLO,KMHI)
      WRITE (6,9300)    ICHNLA(ICHN),(NTBSTO(LM),LM=L1,L7)
 6835 CONTINUE
                                                          GO TO 6900
C
C
C    ******************************************************************
C    *    WAVELENGTH CALIBRATION PERIOD IS BEING PROCESSED             *
C    ******************************************************************
 6850 IPAGE = IPAGE + 1
      MO = NTBSTO(3)
      WRITE (6,9200)    IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
     *                  (NTBSTO(MN),
     *                  MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
     *                  MK=62,65)
      WRITE (6,9305)
      WCAL1 = (NTBSTO(153) * .0001) + FLOAT(NTBSTO(152))
      WCAL2 = (NTBSTO(155) * .0001) + FLOAT(NTBSTO(154))
      WRITE (6,9310) WCAL1,WCAL2,NTBSTO(156)
C
C
 6900 JKTAB = 0
      CALL ZERO(NTBSTO(1),900)
                                                          GO TO 6010
C
C
C    ******************************************************************
C    *    FILE INSERTION ROUTINE                                       *
C    ******************************************************************
 7000 LU = 12
```

```
      LO = 11
      INSERT = 1
      JKTAB = 0
 7010 CALL NTRAN(LU,2,50,NTABUF)
 7020 CALL NTRAN(LU,15,LSTAT)
      IF (LSTAT + 1) 7030,7020,7060
 7030 IF (LSTAT .EQ. -2)                                    GO TO 7050
      WRITE (6,9150) LU,LSTAT
                                                            GO TO 9999
 7050 JKTAB = 0
      IF (NDOLD .EQ. 1)                                     GO TO 6055
                                                            GO TO 6700
C     ******************************************************************
C     *     UPDATE THE RECORD STORAGE COUNTER FOR THE TAB             *
C     ******************************************************************
 7060 JKTAB = JKTAB + 50
 7062 IF (NTABUF(1) .NE. 23)                                GO TO 7064
      KMTAB = 1
                                                            GO TO 7067
 7064 IF (NTABUF(1) .GT. 18)                                GO TO 7065
      KMTAB = ((NTABUF(1) - 1) * 50) + 1
                                                            GO TO 7067
 7065 KMTAB = ((NTABUF(1) - 19)* 50) + 1
C     ******************************************************************
C     *     TRANSFER THE DATA FROM NTABUF TO NTBSTO                   *
C     ******************************************************************
 7067 CALL AMOV(NTABUF(1),NTBSTO(KMTAB),50)
      IF (IIN(3) .GE. 0)
     *                                                      GO TO 7068
      CALL PDUMP(NTBSTO(1),NTBSTO(900),1)
C     ******************************************************************
C     *     CHECK TO SEE IF THE CALIBRATION FILE PERIOD HAS ENDED     *
C     ******************************************************************
 7068 IF (JKTAB .EQ. 600 .AND. NTBSTO(1) .EQ. 1)           GO TO 7070
      IF (JKTAB .EQ. 450 .AND. NTBSTO(1) .EQ. 23)          GO TO 7070
C     ******************************************************************
C     *     CHECK TO SEE IF WAVELENGTH CALIBRATION PERIOD FILE HAS ENDED *
C     ******************************************************************
      IF (JKTAB .EQ. 200 .AND. NTBSTO(1) .EQ. 19)          GO TO 7080
                                                            GO TO 7010
C     ******************************************************************
C     *     WRITE FILE TO NEW HISTORICAL TAPE                          *
C     ******************************************************************
 7070 DO 7077 KTBCAL=1,12
      IF (NTBSTO(1) .EQ. 23 .AND. KTBCAL .GT. 9)           GO TO 7023
      IR = ((KTBCAL - 1)*50) + 1
      IF (NTBSTO(1) .EQ. 23 .AND. KTBCAL .GT. 3) IR=IR+450
      CALL NTRAN(LO,1,50,NTBSTO(IR))
 7074 CALL NTRAN(LO,15,LSTAT)
      IF(LSTAT+1) 7076,7074,7077
 7076 WRITE (6,9050) LO,LSTAT
                                                            GO TO 9999
 7077 CONTINUE
C     ******************************************************************
C     *     CALIBRATION PERIOD IS BEING PROCESSED                     *
C     ******************************************************************
```

FORTRAN V004A   17:35:54   27-JUL-76   PAGE   8

```
7023 IPAGE = IPAGE + 1
     KMLO = 595
     KMHI = 601
     KCHN = 602
     L1 = 607
     MO = NTBSTO(3)
     WRITE (6,9200)     IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
    *                   (NTBSTO(MN),
    *                   MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
    *                   MK=62,65)
     IF (NTBSTO(1) .EQ. 23)                                GO TO 7075
     WRITE (6,9210)
     WRITE (6,9230)
     W( 1) = (NTBSTO(165) * .0001) + FLOAT(NTBSTO(164))
     W( 2) = (NTBSTO(220) * .0001) + FLOAT(NTBSTO(219))
     W( 3) = (NTBSTO(265) * .0001) + FLOAT(NTBSTO(264))
     W( 4) = (NTBSTO(205) * .0001) + FLOAT(NTBSTO(204))
     W( 5) = (NTBSTO(365) * .0001) + FLOAT(NTBSTO(364))
     W( 6) = (NTBSTO(210) * .0001) + FLOAT(NTBSTO(209))
     W( 7) = (NTBSTO(465) * .0001) + FLOAT(NTBSTO(464))
     W( 8) = (NTBSTO(215) * .0001) + FLOAT(NTBSTO(214))
     W( 9) = (NTBSTO(565) * .0001) + FLOAT(NTBSTO(564))
     W(10) = (NTBSTO(305) * .0001) + FLOAT(NTBSTO(304))
     W(11) = (NTBSTO(567) * .0001) + FLOAT(NTBSTO(566))
     W(12) = (NTBSTO(310) * .0001) + FLOAT(NTBSTO(309))
     W(13) = (NTBSTO(315) * .0001) + FLOAT(NTBSTO(314))
     W(14) = (NTBSTO(405) * .0001) + FLOAT(NTBSTO(404))
     W(15) = (NTBSTO(410) * .0001) + FLOAT(NTBSTO(409))
     W(16) = (NTBSTO(415) * .0001) + FLOAT(NTBSTO(414))
     W(17) = (NTBSTO(505) * .0001) + FLOAT(NTBSTO(504))
     W(18) = (NTBSTO(510) * .0001) + FLOAT(NTBSTO(509))
     W(19) = (NTBSTO(515) * .0001) + FLOAT(NTBSTO(514))
     IF (IIN(3) .GE. 0)                                    GO TO 7073
    *
     CALL PDUMP (W(1),W(19),4)
7073 WRITE (6,9240) (W(NK),NK=1,19)
     CALL ZERO(NTBSTO(1),900)
     JKTAB=0
                                                           GO TO 7010

7075 WRITE (6,9220)
     DO 7078 IJ=1,5
     KMLO = KMLO + 7
     KMHI = KMHI + 7
     KCHN = KCHN + 50
     L1 = L1 + 50
     L7 = L1 +  6
     ICHN = NTBSTO(KCHN)
     WRITE (6,9245)
     WRITE (6,9250)     (NTBSTO(KM),KM=KMLO,KMHI)
     WRITE (6,9300)     ICHNLA(ICHN),(NTBSTO(LM),LM=L1,L7)
7078 CONTINUE
C
C
     CALL ZERO(NTBSTO(1),900)
     JKTAB=0
                                                           GO TO 7010
```

```
C     ******************************************************************
C     *    WRITE FILE TO NEW HISTORICAL TAPE                           *
C     ******************************************************************
 7080 DO 7089 KTBWVL=1,4
      IR = ((KTBWVL - 1)*50) + 1
      CALL NTRAN(LO,1,50,NTBSTO(IR))
 7084 CALL NTRAN(LO,15,LSTAT)
      IF(LSTAT+1) 7086,7084,7089
 7086 WRITE (6,9050) LO,LSTAT
                                                    GO TO 9999
 7089 CONTINUE
C     ******************************************************************
C     *    WAVELENGTH CALIBRATION PERIOD IS BEING PROCESSED            *
C     ******************************************************************
 7850 IPAGE = IPAGE + 1
      MO = NTBSTO(3)
      WRITE (6,9200)      IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
     *                    (NTBSTO(MN),
     *                    MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
     *                    MK=62,65)
      WRITE (6,9305)
      WCAL1 = (NTBSTO(153) * .0001) + FLOAT(NTBSTO(152))
      WCAL2 = (NTBSTO(155) * .0001) + FLOAT(NTBSTO(154))
      WRITE (6,9310) WCAL1,WCAL2,NTBSTO(156)
C
C
      CALL ZERO(NTBSTO(1),900)
      JKTAB=0
                                                    GO TO 7010
C
C
C
C
C     ******************************************************************
C     *    TAB OLD HISTORICAL FILE TAPE ONLY                           *
C     ******************************************************************
 8000 LU = 10
      JKTAB=0
      CALL NTRAN(LU,5)
 8010 CALL NTRAN(LU,2,50,NTABUF)
 8020 CALL NTRAN(LU,15,LSTAT)
      IF (LSTAT + 1) 8030,8020,8060
 8030 IF (LSTAT .EQ. -2)                            GO TO 8050
      WRITE (6,9150) LU,LSTAT
                                                    GO TO 9999
 8050 CALL NTRAN(LU,5)
      WRITE (6,9500)
      CALL SEGRET
C     ******************************************************************
C     *    UPDATE THE RECORD STORAGE COUNTER FOR THE TAB               *
C     ******************************************************************
 8060 JKTAB = JKTAB + 50
      NNTAB = JKTAB - 49
      CALL AMOV(NTABUF(1),NTBSTO(NNTAB),50)
C     ******************************************************************
C     *    CHECK TO SEE IF THE CALIBRATION FILE PERIOD HAS ENDED       *
```

```
C     ****************************************************************
      IF (JKTAB .EQ. 600 .AND. NTBSTO(1) .EQ. 1)             GO TO 8070
      IF (JKTAB .EQ. 450 .AND. NTBSTO(1) .EQ. 23)            GO TO 8070
C     ****************************************************************
C     *    CHECK TO SEE IF WAVELENGTH CALIBRATION PERIOD FILE HAS ENDED *
C     ****************************************************************
      IF (JKTAB .EQ. 200 .AND. NTBSTO(1) .EQ. 19)            GO TO 8850
                                                             GO TO 8010
C     ****************************************************************
C     *    CALIBRATION PERIOD IS BEING PROCESSED                       *
C     *    TAB THE CALIBRATION PERIOD                                  *
C     ****************************************************************
 8070 IPAGE = IPAGE + 1
      KMLO = 145
      KMHI = 151
      KCHN = 152
      L1 = 157
      MO = NTBSTO(3)
      WRITE (6,9200)      IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
     *                    (NTBSTO(MN),
     *                    MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
     *                    MK=62,65)
      IF (NTBSTO(1) .EQ. 23)                                  GO TO 8080
      WRITE (6,9210)
      WRITE (6,9230)
      W( 1) = (NTBSTO(165) * .0001) + FLOAT(NTBSTO(164))
      W( 2) = (NTBSTO(220) * .0001) + FLOAT(NTBSTO(219))
      W( 3) = (NTBSTO(265) * .0001) + FLOAT(NTBSTO(264))
      W( 4) = (NTBSTO(205) * .0001) + FLOAT(NTBSTO(204))
      W( 5) = (NTBSTO(365) * .0001) + FLOAT(NTBSTO(364))
      W( 6) = (NTBSTO(210) * .0001) + FLOAT(NTBSTO(209))
      W( 7) = (NTBSTO(465) * .0001) + FLOAT(NTBSTO(464))
      W( 8) = (NTBSTO(215) * .0001) + FLOAT(NTBSTO(214))
      W( 9) = (NTBSTO(565) * .0001) + FLOAT(NTBSTO(564))
      W(10) = (NTBSTO(305) * .0001) + FLOAT(NTBSTO(304))
      W(11) = (NTBSTO(567) * .0001) + FLOAT(NTBSTO(566))
      W(12) = (NTBSTO(310) * .0001) + FLOAT(NTBSTO(309))
      W(13) = (NTBSTO(315) * .0001) + FLOAT(NTBSTO(314))
      W(14) = (NTBSTO(405) * .0001) + FLOAT(NTBSTO(404))
      W(15) = (NTBSTO(410) * .0001) + FLOAT(NTBSTO(409))
      W(16) = (NTBSTO(415) * .0001) + FLOAT(NTBSTO(414))
      W(17) = (NTBSTO(505) * .0001) + FLOAT(NTBSTO(504))
      W(18) = (NTBSTO(510) * .0001) + FLOAT(NTBSTO(509))
      W(19) = (NTBSTO(515) * .0001) + FLOAT(NTBSTO(514))
      WRITE (6,9240) (W(NK),NK=1,19)
      CALL ZERO(NTBSTO(1),900)
      JKTAB=0
                                                              GO TO 8010
 8080 WRITE (6,9220)
      DO 8835 IJ=1,5
      KMLO = KMLO + 7
      KMHI = KMHI + 7
      KCHN = KCHN + 50
      L1 = L1 + 50
      L7 = L1 +  6
      ICHN = NTBSTO(KCHN)
```

```
      WRITE (6,9245)
      WRITE (6,9250)    (NTBSTO(KM),KM=KMLO,KMHI)
      WRITE (6,9300)    ICHNLA(ICHN),(NTBSTO(LM),LM=L1,L7)
 8835 CONTINUE
      CALL ZERO(NTBSTO(1),900)
      JKTAB=0
                                                            GO TO 8010
C
C
C     ***************************************************************
C     *    WAVELENGTH CALIBRATION PERIOD IS BEING PROCESSED            *
C     *    TAB THE WAVELENGTH CALIBRATION PERIOD                       *
C     ***************************************************************
 8850 IPAGE = IPAGE + 1
      MO = NTBSTO(3)
      WRITE (6,9200)    IPAGE,BDATE,NTBSTO(2),RMON(MO),NTBSTO(4),
     *                  (NTBSTO(MN),
     *                  MN=8,15),(NTBSTO(NM),NM=102,106),(NTBSTO(MK),
     *                  MK=62,65)
      WRITE (6,9305)
      WCAL1 = (NTBSTO(153) * .0001) + FLOAT(NTBSTO(152))
      WCAL2 = (NTBSTO(155) * .0001) + FLOAT(NTBSTO(154))
      WRITE (6,9310) WCAL1,WCAL2,NTBSTO(156)
      CALL ZERO(NTBSTO(1),900)
      JKTAB=0
                                                            GO TO 8010
C
C
C     ***************************************************************
C     *                                                              *
C     *    FORMAT STATEMENTS                                         *
C     *                                                              *
C     ***************************************************************
 9050 FORMAT                (1H0,'NTRAN WRITE ERROR ON UNIT',I4,
     *                          'STATUS WORD =',I4)
 9150 FORMAT                (1H0,'NTRAN READ ERROR ON UNIT',I4,
     *                          'STATUS WORD =',I4)
 9200 FORMAT                (1H1,T30,'S 1 9 1 H   P R E P R O C E S S O R   T'
     *                            ,'A P E   Q U A L I T Y   T E S T',T122,
     *                             'PAGE',I4,/,
     *                             T107,'RUN DATE: ',9A1,//,
     *                        T42,'* * * * * H I S T O R I C A L   F I L'
     *                            ,'E * * * * *',//,
     *                        T33,'START DATE: ',I2,'-',A4,I2,T77,'START'
     *                            ,' TIME: ',I2,':',I2,':',I2,'.',I4,/,
     *                        T77,'STOP  TIME: ',I2,':',I2,':',I2,'.',I4,
     *                      //,T33,'MISSION ',I3,T47,'FLIGHT ',I3,T61,'SI'
     *                            ,'TE ',I3,T75,'LINE ',I3,T89,'RUN ',I3,/
     *                       ,T57,'FIRST FRAME TIME:  ',I2,':',I2,':',I2,
     *                            '.',I4,/,
     *                        T57,'----------------------------------'/)
 9210 FORMAT                (1H ,T61,'***CALIBRATION PERIOD***',//)
 9220 FORMAT                (1H ,T59,'***BIAS VOLTAGE HISTOGRAM***',//)
 9230 FORMAT                (1H ,T25,'PARAMETER',T50,'VALUE',T84,'PARAMETER'
     *                       ,T109,'VALUE'/,
     *                        T18,'------------------------',T45,
```

```
                          '---------------',
     *                T77,'------------------------',T104,
     *                  '---------------',/)
9240 FORMAT          (1H ,T18,'AVG. W/L RAMP PEAK VOLT',T45,F8.4,
     *                ' VOLTS',
     *                T77,'SWL CAL LAMP',T102,F10.4,    ' VOLTS',
     *              /,T18,'W/L RAMP PEAK STD. DEV.',T45,F8.4,
     *                ' VOLTS',
     *                T77,'RESPONSIVITY',T93,'(A2)',
     *                T102,F10.4,' V/W-CM2-STR-MICR',
     *              /,T18,'AVG. W/L RAMP MIN. VOLT',T45,F8.4,
     *                ' VOLTS',
     *                T77,'NOISE',T93,'(A2)',
     *                T102,F10.4,     ' VOLTS',/,
     *                T18,'W/L RAMP MIN. STD. DEV.',T45,F8.4,
     *                 ' VOLTS',
     *                T77,'NESR' ,T93,'(A2)',
     *                T102,F10.4,' E-4 W/CM2-STR-MICR',/,
     *                T18,'AVG.  LINEARITY OF SCAN',T45,F8.4,
     *                 ' RMS DEV',
     *                T77,'RESPONSIVITY',T93,'(A3)',
     *                T102,F10.4,' V/W-CM2-STR-MICR',
     *              /,T18,'AVG. SCAN INTERVAL',T45,F8.4,
     *                ' SECONDS',
     *                T77,'NOISE',T93,'(A3)',
     *                T102,F10.4,     ' VOLTS',/,
     *                T77,'NESR' ,T93,'(A3)',
     *                T102,F10.4,' E-4 W/CM2-STR-MICR',/,
     *                T77,'RESPONSIVITY',T93,'(A5)',
     *                T102,F10.4,' V/W-CM2-STR-MICR',/,
     *                T77,'NOISE',T93,'(A5)',
     *                T102,F10.4,     ' VOLTS',/,
     *                T77,'NESR' ,T93,'(A5)',
     *                T102,F10.4,' E-4 W/CM2-STR-MICR',/,
     *                T77,'RESPONSIVITY',T93,'(A6)',
     *                T102,F10.4,' V/W-CM2-STR-MICR',/,
     *                T77,'NOISE',T93,'(A6)',
     *                T102,F10.4,     ' VOLTS',/,
     *                T77,'NESR' ,T93,'(A6)',
     *                T102,F10.4,' E-4 W/CM2-STR-MICR',//)
9245 FORMAT          (1H ,T52,'<OR=',T106,'>OR=')
9250 FORMAT          (1H ,T33,'CHANNEL VALUE',T52,I3,T61,I3,T70.I3,
     *                T79,I3,T88,I3,T97,I3,T106,I3,/,
     *                T33,'------- -----',T52,'---',T61,'---',
     *                T70,'---',T79,'---',T88,'---',T97,'---',
     *                T106,'---')
9300 FORMAT          (1H ,T41,1A2,T51,I4,T60,I4,T69,I4,T78,I4,T87,I4,
     *                T96,I4,T105,I4,//)
9305 FORMAT          (1H ,T61,'***WAVELENGTH CALIBRATION***',//)
9310 FORMAT          (1H ,T33,'CHANNEL A4 PCM VALUES AVG. = ',F8.4,
     *               5X,'STD. DEV. = ',F8.4,      7X,'DATA FROM ',
     *             'CH. A',I1)
9400 FORMAT          (1H1,T30,'S 1 9 1 H   P R E P R O C E S S O R   T'
     *                  ,'A P E   Q U A L I T Y   T E S T',T122,
     *                  'PAGE',I4,/,
     *                   T107,'RUN DATE: ',9A1,//,
```

```
        *                              T52,'* * * Q A  S U M M A R Y * * *',//
        *                             ,T33,'FIRST FRAME TIME:',I2,':',I2,':',F7.4,
        *                              T67,'LAST FRAME TIME:',I2,':',I2,':',F7.4,
        *                           //,T33,'MISSION ',I3,T47,'FLIGHT ',I3,T61,'SI'
        *                                 ,'TE ',I3,T78,'LINE ',I3,T89,'RUN ',I3,/
        *                            /,T52,'***PARAMETER ANOMALY TOTALS***',//,
        *                              T26,'PARAMETER',T46,'TOTAL DETECTED',
        *                              T79,'PARAMETER',T99,'TOTAL DETECTED',/,
        *                              T21,'====================',T46,
        *                              '==============',T74,'===================',
        *                              T99,'==============',///,
        *                              T21,'SCAN LINEARITY',T49,F7.0,
        *                              T74,'SCAN REJECTED',T102,F7.0,/,
        *                              T21,'SYNC PULSE',T49,F7.0,
        *                              T74,'SWL CAL LAMP',T102,F7.0,/,
        *                              T21,'ZERO VOLTS REF.',T49,F7.0,
        *                              T74,'POWER SUPPLY DIAG.',T102,F7.0,/,
        *                              T21,'PACKAGE TEMP.',T49,F7.0,
        *                              T74,'DATA PALLET TEMP.',T102,F7.0,/,
        *                              T21,'LWL DETECTOR TEMP.',T49,F7.0,
        *                              T74,'SPEC. PALLET TEMP.',T102,F7.0,/,
        *                              T21,'DICHROIC TEMP.',T49,F7.0,
        *                              T74,'HEATED CAL TEMP.',T102,F7.0,/,
        *                              T21,'REF. SOURCE TEMP.',T49,F7.0,
        *                              T74,'AMBIENT CAL TEMP.',T102,F7.0,/,
        *                              T21,'INT. SPHERE TEMP.',T49,F7.0,
        *                              T74,'MIRROR TEMP.',T102,F7.0,/,
        *                              T21,'RAD CAL WHEEL POS.',T49,F7.0,
        *                              T74,'FOV FLAG',T102,F7.0,/////,
        *                              T54,'***HISTORICAL FILE FLAGS***',//)
  9450 FORMAT               (1H ,T21,'1.    WAVELENGTH CAL ABORTED')
  9460 FORMAT               (1H ,T21,'2.    RESPONSIVITY ABORTED')
  9465 FORMAT               (1H ,T21,'3.    WAVELENGTH RAMP CALCULATIONS ABOR
      *TED - ZERO GOOD SCANS')
  9470 FORMAT               (1H ,T21,'NO CALCULATIONS ABORTED')
  9500 FORMAT               (1H1)
C     *****************************************************************
  9999 CALL SEGRET
       END

ROUTINES CALLED:
DATE   , AMOV   , STOHMS, IABS   , ZERO   , NTHAN  , PDUMP
FLOAT  , SEGRET  .

SWITCHES = /ON,/SU,/CO

BLOCK        LENGTH
MAIN.    8143    (037636)*
ANDAT    91      (000266)
CALRT    3       (000006)
CMPLET   2       (000004)
DCARGN   28      (000070)
DCDATA   3993    (017462)
ERROR    20      (000050)
HISDAT   8       (000020)
INPUT    158     (000474)
INTNDX   5       (000012)
PRNDX    54      (000154)
QADAT    5       (000012)
TIMES    522     (002024)
TITLES   32      (000100)
```

```
**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES   03750 13516
EXECUTABLES    05805 11461
ASSEMBLY       03828 16355
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:RAMPRO.OBJ,LP:<DK1:RAMPRO.FTN/ON/SU/CO:99
```

```
C     *****************************************************************
C     *                                                               *
C     *                                                               *
C     *     LOAD MODULE : RAMPRO                                      *
C     *                                                               *
C     *                                                               *
C     *****************************************************************
C
C
C
C     *****************************************************************
C     *     COMMON STATEMENTS                                         *
C     *****************************************************************
      COMMON / A4CHNL /  IA4(685)
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *               IWLFG, IRAMFG,LZNE,IPG
      COMMON / CMPLET /  NCMPLT(2)
      COMMON / DCARGN /  START          ,STOP           ,TOL          ,
     1                   IFLAGG         ,FMT            ,IDKRTP       ,
     2                   FRSTR(2)       ,FRSTP(2)       ,IDBLE        ,
     3                   LU             ,ISIZE          ,NTH          ,
     4                   MAX            ,NAV            ,INIT         ,
     5                   IDDC           ,ISTAT
      COMMON / DCDATA /  MEAS(3993)
      COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
     *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
      COMMON / INPUT  /  IIN(102)       ,RIN(28)
      COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
      COMMON / PRNDX  /  PRNDX1(2,9)    ,PRNDX2(2,9)    ,PRNDX3(2,9)
      COMMON / SAVE   /  IBGNSC         ,NMVALD         ,IAHEAD       ,
     1                   IPRVOK         ,ISAVPT         ,IA4NDX       ,
     2                   TMCURR         ,TMFRCR         ,TMFRST       ,
     3                   SUMTM          ,RMSTOT         ,IA6SET       ,
     4                   RMNSUM         ,RMXSUM         ,RXSMSQ       ,
     5                   RNSMSQ
      COMMON / TIMES  /  BJD            ,SSTIME(2,65)
      COMMON / TITLES /  ITITL(32)
C     *****************************************************************
C     *     TYPE STATEMENTS                                           *
C     *****************************************************************
      DOUBLE PRECISION   AVSCTM
      DOUBLE PRECISION   SSTIME         ,START          ,STOP
      DOUBLE PRECISION   FRSPRC         ,LASPRC         ,TMCURR
      DOUBLE PRECISION   TMFRCR         ,TMFRST         ,SUMTM
      DOUBLE PRECISION   TMES69         ,TMS153
      DOUBLE PRECISION   USCNTM         ,TOL
      INTEGER            PRNDX1         ,PRNDX2         ,PRNDX3
      INTEGER            FMT            ,FRSTR          ,FRSTP
      INTEGER            ANSTAT
      INTEGER ANCNTR,ANREC
C     *****************************************************************
C     *     DIMENSION STATEMENTS                                      *
C     *****************************************************************
      DIMENSION IDEVIA(685)
```

```
      DIMENSION NCMPLT(2)
      DIMENSION NTABUF(50)
C     *****************************************************************
C     *    EQUIVALENCE STATEMENTS                                     *
C     *****************************************************************
      EQUIVALENCE        (ICSCN,          IIN(20))
      EQUIVALENCE        (I.TOL,          IIN(21))
      EQUIVALENCE        (I YNC,          IIN(22))
      EQUIVALENCE        (IE ND,          IIN(23))
      EQUIVALENCE        (ISMIN,          IIN(24))
      EQUIVALENCE        (ISMAX,          IIN(25))
C     *****************************************************************
C     *****************************************************************
C
      IF (IIN(3) .GE. 0)
     *                                                 GO TO 4000
      WRITE (6,9000)
      WRITE (6,9200) L
 4000 NX = 0
      CALL AMOV(MEAS(69),TMES69,4)
      CALL AMOV(MEAS(153),TMS153,4)
      IF (IIN(3) .GE. 0)
     *                                                 GO TO 4003
      WRITE (6,9340) TMES69
      WRITE (6,9350) TMS153
C
C
C     *****************************************************************
C     *    CHECK FOR FIRST TIME THROUGH                               *
C     *****************************************************************
 4003 IF (NCMPLT(1) .EQ. 0)                                GO TO 4010
C
C
C     *****************************************************************
C     *    CHECK FOR NEW CAL PERIOD                                   *
C     *****************************************************************
      IF (NCMPLT(1) .NE. L)                                GO TO 4005
      IF (NCMPLT(2) .NE. 1)                                GO TO 4020
      IF (IIN(3) .GE. 0)
     *                                                     GO TO 4004
      ISNO = 4000
      WRITE (6,9050) ISNO
      WRITE (6,9900)
 4004 CALL SEG('DK1:QAEXEC,GGG',0,0)
 4005 IF (NCMPLT(2) .EQ. 1)                                GO TO 4010
      KRET = 2
      JPAR = 20
      NSAM = NCMPLT(1)
      LAN  = 0
      CALL SEG('DK1:ANPRO,LLL',2)
      IF(NMVALD .NE. 0)                                    GO TO 4600
      AVGMX = 0.
      SDPK = 0.
      AVGMN = 0.
      SDMIN = 0.
      AVGLIN = 0.
```

```
      AVSCTM = 0.
                                                          GO TO 4603
C
C
C     *****************************************************************
C     *    EITHER INITIAL PASS OR A NEW CAL PERIOD                    *
C     *****************************************************************
 4010 IA4NDX = 0
      IA6SET = 0
      NCMPLT(1) = L
      NCMPLT(2) = 0
      IBGNSC    = 0
      TMCURR    = 0.
      TMFRST    = 0.
      NMVALD    = 0
      ISAVPT    = 0
 4020 IF (IAHEAD .NE. 1)                                  GO TO 4030
      NX = 1
      IAHEAD = 0
      IF (IIN(3) .GE. 0)
     *                                                    GO TO 4310
      ISNO = 4020
      WRITE (6,9050) ISNO
C
C
C     *****************************************************************
C     *                                                               *
C     *    ENTRY POINT FOR SAMPLE CYCLING                             *
C     *                                                               *
C     *****************************************************************
 4030 IF (IBGNSC .EQ. 0)  IA4NDX = 0
 4040 NX = NX + 1
      IF(MEAS(3399+NX) .LE. ISYNC .AND. NX .NE. 577)   GO TO 4050
      IF (IIN(3) .GE. 0)
     *                                                    GO TO 4050
      ISNO = 4040
      WRITE (6,9050) ISNO
      WRITE (6,9100) NX
      CALL PDUMP(MEAS(3399+NX),MEAS(3399+NX),1)
 4050 IF(NX .LT. PRNDX3(1,1))                             GO TO 4040
C
C
C     *****************************************************************
C     *    CHECK TO SEE IF PROCESSING HAS BEEN COMPLETED FOR THIS MEAS *
C     *    ARRAY                                                      *
C     *****************************************************************
      IF (NX .LE. PRNDX3(2,1))                            GO TO 4060
      NX = NX - 1
      IF (IBGNSC .NE. 0)                                  GO TO 4200
      IF (IIN(3) .GE. 0)
     *                                                    GO TO 4059
      ISNO = 4040
      WRITE (6,9050) ISNO
      WRITE (6,9900)
 4059 CALL SEG('DK1:QAEXEC.GGG',0,0)
 4060 IF (MEAS(3399 + NX) .GT. ISYNC)                     GO TO 4080
```

```
 4061 IA6SET = 0
      IF((IA4NDX + NX - ISAVPT) .LE. ISMAX)           GO TO 4030
      KRET = 2
      JPAR =16
      NSAM =NX
      LAN  = 6
      CALL SEG('DK1:ANPRO.LLL',2)
      CALL ZERO(IA4,685)
      IBGNSC = 0
      IPRVOK = 0
      IA4NDX = 0
      NMVALD = 0
      SUMTM = 0.
      USCNTM = 0.
                                                      GO TO 4030
 4080 IA6SET = IA6SET + 1
      IF (IA6SET .EQ. 2)                              GO TO 4061
      ISAVPT = NX - 1
      IF(NX .LE. 1)                                   GO TO 4085
      IF(MEAS(2246 + NX) .GT. IEND)                   GO TO 4120
 4085 KRET = 2
      JPAR =18
      NSAM =NX
      LAN = 2
      CALL SEG('DK1:ANPRO.LLL',2)
      CALL ZERO(IA4,685)
      IA4NDX = 0
      IPRVOK = 0
      NMVALD = 0
 4120 IF (IIN(3) .GE. 0)
     *                                                GO TO 4125
      ISNO = 4120
      WRITE (6,9050) ISNO
 4125 IF (MEAS(519 + NX) .LE. ISYNC)                  GO TO 4300
      IF (MEAS(1095 + NX).LE. ISYNC)                  GO TO 4300
      IF (MEAS(1671 + NX).LE. ISYNC)                  GO TO 4300
      IF (MEAS(2823 + NX).LE. ISYNC)                  GO TO 4300
                                                      GO TO 4400
C
C
C     ***************************************************************
C     *    PARTIAL LOADING OF IA4 ARRAY                               *
C     ***************************************************************
 4200 IF (IA4NDX .EQ. 0)
     *     TMFRST = TMES69 + (TMS153 * ISAVPT)
      IF (IIN(3) .GE. 0)
     *                                                GO TO 4220
      ISNO = 4200
      WRITE (6,9360) TMFRST
 4220 IPART = NX - ISAVPT
      IPRT = IA4NDX + 1
      CALL AMOV(MEAS(2248 + ISAVPT),IA4(IPRT),IPART)
      IF (IIN(3) .GE. 0)
     *                                                GO TO 4250
      WRITE (6,9050) ISNO
      CALL PDUMP(NX,NX,1)
```

```
      CALL PDUMP(ISAVPT,ISAVPT,1)
      CALL PDUMP(IA4NDX,IA4NDX,1)
      CALL PDUMP(MEAS(2248),MEAS(2823),1)
      WRITE (6,9050) ISNO
      CALL PDUMP(IA4(1),IA4(685),1)
C
C
C     ******************************************************************
C     *    TERMINATES PROCESSING FOR THE PRESENT MEAS ARRAY            *
C     ******************************************************************
 4250 IA4NDX = IA4NDX + IPART
      ISAVPT = 0
      IAHEAD = 0
      IF (IIN(3) .GE. 0)
     *                                                   GO TO 4260
      WRITE (6,9050) ISNO
      WRITE (6,9900)
 4260 CALL SEG('DK1:QAEXEC.GGG',0,0)
C
C
C     ******************************************************************
C     *    PROCESSING DATA THAT IS PART OF A VALID SCAN LINE           *
C     ******************************************************************
 4300 IF (IIN(3) .GE. 0)
     *                                                   GO TO 4305
      ISNO = 4300
      WRITE (6,9050) ISNO
      CALL PDUMP(PRNDX3(2,1),PRNDX3(2,1),1)
 4305 IF((NX + 1) .LE. PRNDX3(2,1))                      GO TO 4310
      IAHEAD = 1
                                                         GO TO 4200
 4310 IF (MEAS(520 + NX) .GT. ISYNC)                     GO TO 4320
      LAN = 1
                                                         GO TO 4380
 4320 IF (MEAS(1096 +NX) .GT. ISYNC)                     GO TO 4330
      LAN = 2
                                                         GO TO 4380
 4330 IF (MEAS(1672 +NX) .GT. ISYNC)                     GO TO 4340
      LAN = 3
                                                         GO TO 4380
 4340 IF (MEAS(2824 +NX) .GT. ISYNC)                     GO TO 4400
      LAN  = 2
C
C
C     ******************************************************************
C     *    BAD SYNC PULSE                                              *
C     ******************************************************************
 4380 KRET = 2
      NSAM =NX
      JPAR =16
      CALL SEG('DK1:ANPRO.LLL',2)
C
C
C     ******************************************************************
C     ******************************************************************
C
```

```
C
 4400 IF (IBGNSC .EQ. 0)                                        GO TO 4420
 4405 ITOTAL = IA4NDX + ISAVPT
      IF(ITOTAL .GT. ISMAX .OR. ITOTAL .LT. ISMIN)              GO TO 4410
C
C
C    ******************************************************************
C    *    PREVIOUS SCAN IS GOOD                                       *
C    ******************************************************************
      IPRVOK = 1
                                                                GO TO 4420
C
C
C    ******************************************************************
C    *    PREVIOUS SCAN IS EITHER TOO LONG OR TOO SHORT               *
C    *    ANPRO OUTPUTS APPROPRIATE MESSAGE                           *
C    ******************************************************************
 4410 KRET = 2
      JPAR =18
      NSAM =NX
      LAN  = ITOTAL
      CALL SEG('DK1:ANPRO.LLL',2)
      CALL ZERO(IA4,685)
      ISAVPT = NX - 1
      IA4NDX = 0
      IPRVOK = 0
      NMVALD = 0
C
C
C    ******************************************************************
C    *    CHECK FOR VALID START FOR NEXT SCAN LINE                    *
C    ******************************************************************
 4420 IF (MEAS(2248 + NX) .LT. 50)                              GO TO 4460
C
C
C    ******************************************************************
C    *    BAD START OF SCAN DETECTED                                  *
C    ******************************************************************
      IBGNSC = 0
      KRET   = 2
      JPAR   =18
      NSAM   =NX
      LAN    = 1
      CALL SEG('DK1:ANPRO.LLL',2)
      NMVALD = 0
                                                                GO TO 4480
 4460 IBGNSC = 1
 4480 IF (IPRVOK .NE. 1)                                        GO TO 4030
C
C
C    ******************************************************************
C    *    MOVE FINAL PART OF PREVIOUS SCAN                            *
C    ******************************************************************
      IF (IIN(3) .GE. 0)
     *                                                          GO TO 4483
      ISNO = 4460
```

```
      WRITE (6,9050) ISNO
 4483 IF(NX .EQ. 1)                                        GO TO 4485
      IPRT  = IA4NDX + 1
      IF (IIN(3) .GE. 0)
     *                                                     GO TO 4484
      CALL PDUMP(IPRT,IPRT,1)
      CALL PDUMP(ISAVPT,ISAVPT,1)
      CALL PDUMP(NX,NX,1)
      CALL PDUMP(IA4NDX,IA4NDX,1)
      CALL PDUMP(MEAS(2248),MEAS(2823),1)
      WRITE (6,9050) ISNO
 4484 CALL AMOV(MEAS(2248),IA4(IPRT),ISAVPT)
      IF (IIN(3) .GE. 0)
     *                                                     GO TO 4485
      CALL PDUMP(IA4(1),IA4(685),1)
 4485 ISAVPT = NX - 1
C
C
C
C
C
C     *****************************************************************
C     *****************************************************************
C     *****************************************************************
C     *    PROCESS THE SCAN LINE                                      *
C     *    IRMPST-LOCATION OF THE FIRST VALUE IN THE ARRAY GE 30      *
C     *****************************************************************
      IRMPST = 0
C
C
C     *****************************************************************
C     *    DETERMINE FIRST LOCATION WHERE A4 IS GREATER THAN OR EQUAL *
C     *    TO 30 PCM COUNTS                                           *
C     *    ( EXCLUDING LOCATION 1,SINCE IT MAY HAVE A VALUE GT 800)   *
C     *****************************************************************
      DO 4500 NGT=2,IA4NDX
      IF (IA4(NGT) .LT. 30)                                GO TO 4500
      IRMPST = NGT
                                                           GO TO 4510
 4500 CONTINUE
C
C
C     *****************************************************************
C     *    ACCUMULATORS FOR INDIVIDUAL SCAN LINE TOTALS               *
C     *****************************************************************
 4510 IF (IIN(3) .GE. 0)
     *                                                     GO TO 4515
      CALL PDUMP(IRMPST,IRMPST,1)
 4515 IA4NDX = IA4NDX + ISAVPT
      IDEC  = IA4NDX + 1
      SUMX  = 0.
      SUMY  = 0.
      SUMXY = 0.
      MXA4  = 0
      MNA4  = IA4(2)
      SMXSQ = 0.
```

```
      SMYSQ = 0.
      IF (NMVALD .NE. 0)                                   GO TO 4530
C
C
C     ******************************************************************
C     *    ZERO OVERALL ACCUMULATORS                                   *
C     ******************************************************************
      SUMTM  = 0
      RMSTOT = 0.
      RMXSUM = 0.
      RMNSUM = 0.
      RXSMSQ = 0.
      RNSMSQ = 0.
C
C
C     ******************************************************************
C     *    DETERMINE PEAK AND MINIMUM VALUES OF A4 FOR THE SCAN        *
C     ******************************************************************
 4530 DO 4550 L4=1,IA4NDX
      IDEC = IDEC - 1
      IF (IA4(IDEC) .GT. MXA4) MXA4 = IA4(IDEC)
      IF (IA4(IDEC) .LE. MNA4) MNA4 = IA4(IDEC)
      IF (IDEC .LT. IRMPST)                                GO TO 4550
      SUMX = SUMX + IDEC
      SUMY = SUMY + IA4(IDEC)
      SUMXY= SUMXY + (FLOAT(IDEC) * IA4(IDEC))
      SMXSQ= SMXSQ + (FLOAT(IDEC)**2)
      SMYSQ= SMYSQ + (FLOAT(IA4(IDEC))**2)
 4550 CONTINUE
C
C
C     ******************************************************************
C     *    COMPUTE STRAIGHT LINE EQUATION : Y = B0 + B1*X              *
C     ******************************************************************
C     ******************************************************************
C     *    COMPUTE B1 COEFFICIENT                                      *
C     ******************************************************************
 4570 IF (IIN(3) .GE. 0)
     *                                                     GO TO 4575
      CALL PDUMP(SUMX,SUMX,4)
      CALL PDUMP(SUMY,SUMY,4)
      CALL PDUMP(SUMXY,SUMXY,4)
      CALL PDUMP(SMXSQ,SMXSQ,4)
      CALL PDUMP(SMYSQ,SMYSQ,4)
 4575 RA4NDX = IA4NDX - IRMPST + 1
      B1 = ((RA4NDX*SUMXY) - (SUMX * SUMY)) /
     *      ((RA4NDX*SMXSQ) - (SUMX ** 2))
      IF (IIN(3) .GE. 0)
     *                                                     GO TO 4580
      CALL PDUMP(B1,B1,4)
C
C
C     ******************************************************************
C     *    COMPUTE B0 COEFFICIENT                                      *
C     ******************************************************************
 4580 BZRO = (SUMY/RA4NDX) - (B1 * (SUMX/RA4NDX))
```

```
      IF (IIN(3) .GE. 0)
     *                                                  GO TO 4585
      CALL PDUMP(BZRO,BZRO,4)
 4585 RMSUM = 0.
C
C
C     ******************************************************************
C     *    DETERMINE DEVIATION OF ACTUAL VALUES FROM ESTIMATED VALUES  *
C     ******************************************************************
      DO 4590 IDEV=IRMPST,IA4NDX
      YEST = BZRO + (FLOAT(IDEV) * B1)
      YACT = IA4(IDEV)
      IF ( ABS(YACT  - YEST) .GT. ILTOL)                GO TO 4650
C
C
C     ******************************************************************
C     *    COMPUTE THE SQUARE OF THE INDIVIDUAL DEVIATION              *
C     *    ACCUMULATE THE SUM OF THE SQUARE OF DEVIATIONS              *
C     ******************************************************************
      RMSIND = (YACT - YEST)**2
      RMSUM  = RMSUM + RMSIND
 4590 CONTINUE
      IF (IIN(3) .GE. 0)
     *                                                  GO TO 4592
      WRITE (6,9320) RMSUM
C
C
C     ******************************************************************
C     *    COMPUTE RMS FOR THE SCAN LINE                               *
C     ******************************************************************
 4592     RMSFNL = SQRT(RMSUM / RA4NDX)
      IF (IIN(3) .GE. 0)
     *                                                  GO TO 4594
      CALL PDUMP(RMSFNL,RMSFNL,4)
C
C
C     ******************************************************************
C     *    COMPUTE THE SCAN LINE TIME                                  *
C     ******************************************************************
 4594 TMCURR = TMES69 + (TMS153 * (NX - 2))
      USCNTM = TMCURR - TMFRST
      IF (IIN(3) .GE. 0)
     *                                                  GO TO 4596
      WRITE (6,9360) TMFRST
      WRITE (6,9370) TMCURR
      CALL PDUMP (USCNTM,USCNTM,5)
C
C
C     ******************************************************************
C     *    UPDATE THE TOTAL ACCUMULATORS                               *
C     ******************************************************************
 4596     SUMTM  = SUMTM + USCNTM
          RMSTOT = RMSTOT + RMSFNL
          RMXSUM = RMXSUM + MXA4
          RMNSUM = RMNSUM + MNA4
          RXSMSQ = RXSMSQ + FLOAT(MXA4)**2
```

```
           RNSMSQ = RNSMSQ + FLOAT(MNA4)**2
        IF (IIN(3) .GE. 0)
       *                                                  GO TO 4598
        WRITE (6,9330)
        CALL PDUMP(SUMTM,SUMTM,5)
        CALL PDUMP(RMSTOT,RMSTOT,4)
        CALL PDUMP(RMXSUM,RMXSUM,4)
        CALL PDUMP(RMNSUM,RMNSUM,4)
        CALL PDUMP(RXSMSQ,RXSMSQ,4)
        CALL PDUMP(RNSMSQ,RNSMSQ,4)
        CALL PDUMP(MXA4,MXA4,1)
        CALL PDUMP(MNA4,MNA4,1)
C
C
C     ****************************************************************
C     *    UPDATE VALID SCAN LINE COUNTER                            *
C     ****************************************************************
 4598      NMVALD = NMVALD + 1
        IF (NMVALD .NE. ICSCN)                            GO TO 4800
C
C
C
C
C     ****************************************************************
C     *    PROCESSING FOR THE REQUESTED NO. OF SCANS HAS BEEN COMPLETED *
C     ****************************************************************
C
C
C
C     ****************************************************************
C     *    COMPUTE AVERAGE OF THE SCAN LINE PEAK VALUES              *
C     *    COMPUTE AVERAGE OF THE SCAN LINE MINIMUM VALUES           *
C     *    COMPUTE STANDARD DEVIATION OF THE SCAN LINE PEAK VALUES   *
C     *    COMPUTE STANDARD DEVIATION OF THE SCAN LINE MINIMUM VALUES *
C     *    COMPUTE AVERAGE OF THE SCAN LINE'S ROOT MEAN SQUARE VALUES *
C     *    COMPUTE AVERAGE TIME OF A SCAN LINE                       *
C     ****************************************************************
 4600 AVGMX = (RMXSUM / NMVALD) * .005
      AVGMN = (RMNSUM / NMVALD) * .005
      IF (NMVALD .NE. 1)                                  GO TO 4601
      SDPK = 0.
      SDMIN = 0.
                                                          GO TO 4602
 4601 SDPK    =.005 * SQRT(((RXSMSQ * NMVALD) - (RMXSUM**2))/
     *          (NMVALD*(NMVALD - 1)))
      SDMIN   =.005 * SQRT(((RNSMSQ * NMVALD) - (RMNSUM**2))/
     *          (NMVALD*(NMVALD - 1)))
 4602 AVGLIN = (RMSTOT/NMVALD)*.005
      AVSCTM = (SUMTM/NMVALD)
C
C
C
C     ****************************************************************
C     *    OUTPUT :                                                  *
C     *    AVG W/L RAMP PEAK       (AVGMX )                          *
C     *    AVG W/L RAMP MIN        (AVGMN )                          *
```

```
C     *    SD  W/L RAMP PEAK        (SDPK  )                           *
C     *    SD  W/L RAMP MIN         (SDMIN )                           *
C     *    AVG SCAN RMS             (AVGLIN)                           *
C     *    AVG SCAN INTERVAL TIME   (AVSCTM)                           *
C     ******************************************************************
      IF (IIN(3) .GE. 0)                                 GO TO 4603
     *
      CALL PDUMP(AVGMX,AVGMX,4)
      CALL PDUMP(AVGMN,AVGMN,4)
      CALL PDUMP(SDPK,SDPK,4)
      CALL PDUMP(SDMIN,SDMIN,4)
      CALL PDUMP(AVGLIN,AVGLIN,4)
      CALL PDUMP(AVSCTM,AVSCTM,5)
 4603 LO = 12
      INTNUM = AVGMX
      RFRAC = AVGMX - INTNUM
      IFRAC = RFRAC * 10000.
      CALL ZERO(NTABUF(1),50)
C
C
C     ******************************************************************
C     *    OUTPUT RECORD TYPE 4                                        *
C     ******************************************************************
      NTABUF(1) = 4
      NTABUF(14)= INTNUM
      NTABUF(15)=IFRAC
 4605 CALL NTRAN(LO,1,50,NTABUF(1))
 4607 CALL NTRAN(LO,15,LSTAT)
      IF (LSTAT + 1) 4609,4607,4611
 4609 WRITE(6,9030) LO,LSTAT
      CALL SEGRET
 4611 INTNUM = SDPK
      RFRAC = SDPK  - INTNUM
      IFRAC = RFRAC * 10000.
      CALL ZERO(NTABUF(1),50)
C
C
C     ******************************************************************
C     *    OUTPUT RECORD TYPE 6                                        *
C     ******************************************************************
      NTABUF(1) = 6
      NTABUF(14)= INTNUM
      NTABUF(15)=IFRAC
 4615 CALL NTRAN(LO,1,50,NTABUF(1))
 4617 CALL NTRAN(LO,15,LSTAT)
      IF (LSTAT + 1) 4619,4617,4621
 4619 WRITE(6,9030) LO,LSTAT
      CALL SEGRET
 4621 INTNUM = AVGMN
      RFRAC = AVGMN - INTNUM
      IFRAC = RFRAC * 10000.
      CALL ZERO(NTABUF(1),50)
C
C
C     ******************************************************************
C     *    OUTPUT RECORD TYPE 8                                        *
```

```
C     ******************************************************************
      NTABUF(1) = 8
      NTABUF(14)= INTNUM
      NTABUF(15)=IFRAC
 4625 CALL NTRAN(LO,1,50,NTABUF(1))
 4627 CALL NTRAN(LO,15,LSTAT)
      IF (LSTAT + 1) 4629,4627,4631
 4629 WRITE(6,9030) LO,LSTAT
      CALL SEGRET
 4631 INTNUM = SDMIN
      RFRAC = SDMIN - INTNUM
      IFRAC = RFRAC * 10000.
      CALL ZERO(NTABUF(1),50)
C
C
C     ******************************************************************
C     *    OUTPUT RECORD TYPE 10                                       *
C     ******************************************************************
      NTABUF(1) = 10
      NTABUF(14)= INTNUM
      NTABUF(15)=IFRAC
 4635 CALL NTRAN(LO,1,50,NTABUF(1))
 4637 CALL NTRAN(LO,15,LSTAT)
      IF (LSTAT + 1) 4639,4637,4641
 4639 WRITE(6,9030) LO,LSTAT
      CALL SEGRET
 4641 INTNUM = AVGLIN
      RFRAC = AVGLIN - INTNUM
      IFRAC = RFRAC * 10000.
      JNTNUM = AVSCTM
      XFRAC = AVSCTM - JNTNUM
      JFRAC = XFRAC * 10000.
      CALL ZERO(NTABUF(1),50)
C
C
C     ******************************************************************
C     *    OUTPUT RECORD TYPE 12                                       *
C     ******************************************************************
      NTABUF(1) = 12
      NTABUF(14)= INTNUM
      NTABUF(15)=IFRAC
      NTABUF(16)= JNTNUM
      NTABUF(17)=JFRAC
 4645 CALL NTRAN(LO,1,50,NTABUF(1))
 4647 CALL NTRAN(LO,15,LSTAT)
      IF (LSTAT + 1) 4648,4647,4649
 4648 WRITE(6,9030) LO,LSTAT
      CALL SEGRET
C
C
C
C     ******************************************************************
C     *    ZERO COUNTERS IN PREPARATION FOR ANOTHER CAL PERIOD         *
C     ******************************************************************
 4649 SUMTM = 0.
      NCMPLT(2) = 1
```

```
      NMVALD = 0
      RMXSUM = 0.
      RMNSUM = 0.
      RMSTOT = 0.
      RXSMSQ = 0.
      RNSMSQ = 0.
      CALL ZERO(IA4,685)
      IDONE = IDONE + 5
      IRMFLG = 1
      CALL SEG('DK1:QAEXEC.GGG',0,0)
C
C
C
C     ******************************************************************
C     *    REJECT SCAN DUE TO DATA OUT OF TOLERANCE                    *
C     ******************************************************************
 4650 KRET = 2
      JPAR =18
      IADJ = IA4NDX - NX + 1
      NSAM = IDEV - IADJ
      LAN  = 9
      IF (IIN(3) .GE. 0)
     *                                                 GO TO 4655
      ISNO = 4650
      WRITE (6,9050) ISNO
      CALL PDUMP(YEST,YEST,4)
      CALL PDUMP(YACT,YACT,4)
      CALL PDUMP(IDEV,IDEV,1)
 4655 CALL SEG('DK1:ANPRO.LLL',2)
      IPRVOK = 0
      NMVALD = 0
 4800 IF (IIN(3) .GE. 0)
     *                                                 GO TO 4810
      ISNO = 4800
      WRITE (6,9050) ISNO
      WRITE (6,9300) IA6SET
      WRITE (6,9310) IBGNSC
 4810 CALL ZERO(IA4,685)
      IA4NDX =0
      TMFRST =0.
      TMCURR =0.
      IF (NX .LE. PRNDX3(2,1))                         GO TO 4030
      IF (IIN(3) .GE. 0)
     *                                                 GO TO 4820
      ISNO = 4800
      WRITE (6,9050) ISNO
      WRITE (6,9900)
 4820 CALL SEG('DK1:QAEXEC.GGG',0,0)
C
C
C
C
C     ******************************************************************
C     *    FORMAT STATEMENTS                                           *
C     ******************************************************************
 9000 FORMAT (1X,'ENTERING RAMPRO ')
```

```
 9030 FORMAT              (1H0,'NTRAN WRITE ERROR ON UNIT',I4,
     *                     'STATUS WORD =',I4)
 9050 FORMAT (1X,'STATEMENT NUMBER ',I5)
 9100 FORMAT (1X,'SAMPLE NUMBER ',I5)
 9200 FORMAT (1X,'DATA VALUE:',I5)
 9300 FORMAT (1X,'IA6SET:',I5)
 9310 FORMAT (1X,'IBGNSC:',I5)
 9320 FORMAT (1X,'RMSUM:',F15.7)
 9330 FORMAT (1X,'TOTAL ACCUMULATORS')
 9340 FORMAT (1X,'START TIME OF MEAS ARRAY',D22.14)
 9350 FORMAT (1X,'DELTA TIME FOR A4   ',D22.14)
 9360 FORMAT (1X,'START TIME FOR THIS SCAN   TMFRST:',D22.14)
 9370 FORMAT (1X,'STOP TIME FOR THIS SCAN     TMCURR:',D22.14)
 9900 FORMAT (1X,'LEAVING RAMPRO ')
      END

ROUTINES CALLED:
AMOV  , SEG   , PDUMP , ZERO  , FLOAT , ABS   , SQRT
NTRAN , SEGRET

SWITCHES = /ON,/SU,/CO

BLOCK      LENGTH
MAIN.   4667   (022166)*
A4CHNL  685    (002532)
ANDAT   91     (000266)
CMPLET  2      (000004)
DCARGN  28     (000070)
DCDATA  3993   (017462)
HISDAT  8      (000020)
INPUT   158    (000474)
INTNDX  5      (000012)
PRNDX   54     (000154)
SAVE    33     (000102)
TIMES   522    (002024)
TITLES  32     (000100)

**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES 04778 12488
EXECUTABLES  05511 11755
ASSEMBLY     03415 16768
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:RESPRO.OBJ,LP:<DK1:RESPRO.FTN/ON/SU/CO:99
```

```
C
C      *****************************************************************
C      *                                                               *
C      *    LOAD MODULE: RESPRO                                        *
C      *                                                               *
C      *****************************************************************
C
C      *****************************************************************
C      *    RESPRO COMPUTES RESPONSIVITY FOR SWL AND HEATED CAL PERIODS  *
C      *****************************************************************
       DOUBLE PRECISION RTME,SSTIME,DSEC
       DOUBLE PRECISION START,STOP,TOL
       REAL IHT,  IRT,  IDT
       REAL CNVTMP(60)
       INTEGER  ANSTAT
       INTEGER ANCNTR,ANREC
       INTEGER PRNDX1,PRNDX2,PRNDX3
       INTEGER FMT,FRSTR,FRSTP
       DIMENSION JCMPLT(2),KCMPLT(2),NA4(685),ITIM(2),NTABUF(50)
       COMMON /PRNDX/ PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
       COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
      *              ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
      *              IWLFG, IRAMFG,LZNE,IPG
       COMMON / DCDATA /  MEAS(3993)
       COMMON /DCARGN/ START,  STOP,      TOL,       IFLAGG,
      1                FMT,    IDKRTP,    FRSTR(2),  FRSTP(2),
      2                IDBLE,  LU,        ISIZE,     NTH,
      3                MAX,    NAV,       INIT,      IDDC,
      4                ISTAT
       COMMON /RESDAT/ JCMPLT(2),KCMPLT(2),NA4(685),NFSCN,NX,NSAMPL,NIX
      *,AVJ2, AVJ3, AVJ5, AVJ6, SVJ2, SVJ3, SVJ5, SVJ6, JSCN6, NSSCAN
      *,    IPSOK
       COMMON  /SIXSV/  ISIX
       COMMON / CALRT /  JJNT,                NDICAL,              NTRCAL
       COMMON /HISDAT/   IRMFLG,    IRSFLG,   IBVFLG,    IWLFLG,
      *                  ICLHDR,    IWLHDR,   IDONE,     IBVHDR
       COMMON /INPUT/ IIN(102),RIN(28)
       COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
       COMMON /RECPTR/ IVAR2
       COMMON /TIMES/ BJD, SSTIME(2,65)
C
C
C
       EQUIVALENCE (IIN(22),ISYNC),(IIN(23),IEND),(IIN(24),ISMIN),
      1            (IIN(25),ISMAX)
       DATA EP/2.7182818284/
       DATA CNVTMP/.153262E+2,.359011E-1,.648069E-5,.882920E-8,
      *.121789E-10,.0E+0,.456401E+1,.557554E-1,4*.0E+0,.437654E+1,
      *.557587E-1,4*.0E+0,.437654E+1,.557587E-1,4*.0E+0,.317003E+0,
      *.431506E-1,-.156785E-4,.198965E-7,-.117920E-10,.451193E-14,
      *-.469973E+1,.695355E-1,-.524248E-4,.928606E-7,-.853239E-10,
      *.407347E-13,-.289909E+2,.160468E+0,-.310689E-3,.516876E-6,
      *-.436491E-9,.154279E-12,.426162E+1,.520435E-1,-.237950E-4,
      *.525359E-7,-.511638E-10,.267007E-13,.432203E+1,.449168E-1,
      *-.140419E-4,.244692E-7,-.184496E-10,.929381E-14,.610345E+2,
```

```
      *,862490E-1,4*,0E+0/
       IF(ISTAT .GT. 0 .AND. IRSFLG .EQ. 1)                       GOTO 2955
       IF(PRNDX3(1,7) .GT. 0)GOTO 2050
       IF(PRNDX3(1,8) .LE. 0)GOTO 2999
C      ** JCMPLT IS COMPLETION FLAG FOR SWL CAL PERIOD **
C
       IF(JCMPLT(1) .EQ. 0)GOTO 2010
       IF(JCMPLT(1) .EQ. LSWL)
       IF(JCMPLT(2) .EQ. 1)GOTO 2010                              GOTO 2020
C
C      ANOMALOUS CONDITION - NOT ENOUGH SWL CAL SCANS
C
       JPAR=17
       LAN=0
       NSAM=0
       KRET=1
       CALL SEG('DK1:ANPRO.LLL',2)
C

C                                                                  GOTO 2960
C
C      SET UP SWL CAL INTERVAL START TIME (L)
C
 2010  JCMPLT(1)=LSWL
       JCMPLT(2)=0
       GOTO 2030
C
C      SWL CAL PERIOD COMPLETE WHEN JCMPLT(2) = 1
C
 2020  IF(JCMPLT(2) .EQ. 1)GOTO 2999
C
C      **  ILO=START INDEX  IHI=STOP INDEX IPRO=0 MEANS SWL PROCESSING **
C
 2030  ILO=PRNDX3(1,8)
       IHI=PRNDX3(2,8)
       IPRO=0
       GOTO 2100
C      **  KCMPLT IS COMPLETION FLAG FOR HEATED CAL PERIOD **
C
 2050  IF(KCMPLT(1) .EQ. 0)GOTO 2060
       IF(KCMPLT(1) .EQ. LHEAT)
       IF(KCMPLT(2) .EQ. 1)GOTO 2060                              GOTO 2070
C
C      ** ANOMALOUS CONDITION - NOT ENOUGH HEATED CAL SCANS **
C
       JPAR=17
       LAN=1
       NSAM=0
       KRET=1
       CALL SEG('DK1:ANPRO.LLL',2)
C

C                                                                  GOTO 2976
C
C      ** SET UP HEATED CAL INTERVAL START TIME (L)
C
 2060  KCMPLT(1)=LHEAT
       KCMPLT(2)=0
```

```
      GOTO 2080
C
C     ** HEATED CAL PERIOD COMPLETE WHEN KCMPLT(2) = 1 **
C
 2070 IF(KCMPLT(2) .EQ. 1)GOTO 2999
C
C     ** ILO=START  IHI=STOP  WHEN IPRO=1 HEATED CAL PROCESSING **
C
 2080 ILO=PRNDX3(1,7)
      IHI=PRNDX3(2,7)
      IPRO=1
C
C     ** START SCAN START/STOP DETECTION **
C     ** NFSCN = 1 SAYS NOT LOOKING FOR VERY FIRST SCAN **
C
 2100 IF(NFSCN .EQ. 1)GOTO 2110
      NSAMPL=1
C
C     ** SET UP MEAS ARRAY LOCATORS FOR ALL SIX CHANNELS **
C
 2110 ILL=ILO-1
      LOCA1=520+ILL
      LOCA2=1096+ILL
      LOCA3=1672+ILL
      LOCA4=2248+ILL
      LOCA5=2824+ILL
      LOCA6=3400+ILL
C
C     ** NX IS THE SCAN SAMPLE COUNTER **
C
 2120 NX=0
      IF(ISTA .EQ. 1)                                       GOTO 2200
C
C     ** CHECK CHANNEL A6 FOR SYNC PULSE (ISYNC) **
C
 2130 IF(MEAS(LOCA6+NX) .LT. ISYNC)GOTO 2160
      ISIX=ISIX+1
      IF(ISIX .EQ. 2)GOTO 2160
      IF((NX-1) .LT. 0)                                     GOTO 2170
C
C     ** CHECK CHANNEL A4 FOR MINIMUM PEAK COUNT (IEND) **
C
      IF(MEAS(LOCA4+(NX-1)) .LT. IEND)                      GOTO 2700
C
C     ** CHECK SYNC PULSES ON OTHER CHANNELS **
C
 2140 IF(MEAS(LOCA1+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA2+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA3+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA5+NX) .LT. ISYNC)GOTO 2190
      GOTO 2500
C     ** CHECK FOR SCAN TOO LONG (> ISMAX) **
C
 2160 ISIX=0
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2300
      ITST=ILO+NX+NIX-1
```

```
      IF(ITST .GE. IHI)                                     GOTO 2180
 2170 NX=NX+1
      GOTO 2130
C
C
 2180 IF(NFSCN .EQ.0)GOTO 2999
C
C     ** MOVE PART OF SCAN TO CH. A4 ARRAY (NA4)
C
      READ(2'1)NA4
      CALL AMOV(MEAS(LOCA2),NA4(NSAMPL),NX)
      WRITE(2'1)NA4
      READ(2'2)NA4
      CALL AMOV(MEAS(LOCA3),NA4(NSAMPL),NX)
      WRITE(2'2)NA4
      READ(2'3)NA4
      CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      WRITE(2'3)NA4
      READ(2'4)NA4
      CALL AMOV(MEAS(LOCA5),NA4(NSAMPL),NX)
      WRITE(2'4)NA4
      READ(2'5)NA4
      CALL AMOV(MEAS(LOCA6),NA4(NSAMPL),NX)
      WRITE(2'5)NA4
      NSAMPL=NSAMPL+NX
      NFSCN=1
      NIX=0
      GOTO 2999
C
C     ** CHECK FOR END OF MEAS ARRAY OR DATA PERIOD **
C
 2190 IF((NX+1) .GT. IHI)GOTO 2400
      II=NX+1
      GOTO 2210
 2200 II=NX
C
C     ** CHECK NEXT VALUES IN OTHER CHANNELS FOR SYNC PULSES **
C
 2210 IF(MEAS(LOCA1+II) .LT. ISYNC)GOTO 2221
      IF(MEAS(LOCA2+II) .LT. ISYNC)GOTO 2222
      IF(MEAS(LOCA3+II) .LT. ISYNC)GOTO 2223
      IF(MEAS(LOCA5+II) .LT. ISYNC)GOTO 2225
      GOTO 2500
C
C     ** ANOMALY - BAD SYNC PULSE -- SCAN NOT REJECTED **
C
 2221 LAN=1
      GOTO 2230
 2222 LAN=2
      GOTO 2230
 2223 LAN=3
      GOTO 2230
 2225 LAN=5
 2230 JPAR=16
      NSAM=NX
      KRET=1
```

```
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2500
C
C     ** PREVIOUS SCAN INVALID -- ZERO NA4 ARRAY **
C
 2300 CALL ZERO(NA4(1),685)
      DO 2350 I=1,5
      WRITE(2'I)NA4
 2350 CONTINUE
      NSAMPL=1
C     ** ANOMALY -- BAD SYNC PULSE ON CH. A6 **
      NFSCN=1
      JPAR=16
      NSAM=NX
      LAN=6
      KRET=1
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2170
C
C     ** END OF MEAS ARRAY -- MOVE GOOD DATA TO NA4 ARRAY **
C
 2400 ISTA=1
      READ(2'1)NA4
      CALL AMOV(MEAS(LOCA2),NA4(NSAMPL),NX)
      WRITE(2'1)NA4
      READ(2'2)NA4
      CALL AMOV(MEAS(LOCA3),NA4(NSAMPL),NX)
      WRITE(2'2)NA4
      READ(2'3)NA4
      CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      WRITE(2'3)NA4
      READ(2'4)NA4
      CALL AMOV(MEAS(LOCA5),NA4(NSAMPL),NX)
      WRITE(2'4)NA4
      READ(2'5)NA4
      CALL AMOV(MEAS(LOCA6),NA4(NSAMPL),NX)
      WRITE(2'5)NA4
      GOTO 2999
C
C     ** CHECK FOR CORRECT NUMBER OF SCANS **
C
 2500 IF(NFSCN .EQ. 0)                                 GOTO 2530
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2510
      IF((NSAMPL+NX) .LT. ISMIN)GOTO 2510
      IPSOK=1
      GOTO 2530
C
C     ** ANOMALY -- SCAN TOO LONG OR TOO SHORT **
C
 2510 JPAR=18
      NSAM=NX
      IPSOK=0
      NFSCN=0
      LAN=NSAMPL+NX
      KRET=1
      CALL SEG('DK1:ANPRO.LLL',2)
```

```
      IHI=IHI-NIX
C
C     ** CHECK FOR FIRST CH. A4 SAMPLE LESS THAN 50 PCM CNTS **
C
 2530 II=LOCA4+NX+1
      IF(MEAS(II) .LT. 50)GOTO 2560
      NFSCN=0
C
C     ** ANOMALY -- FIRST SAMPLE IN A4 TOO HIGH **
C
      JPAR=18
      NSAM=NX
      LAN=2
      KRET=1
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2570
 2560 NFSCN=1
C
C     ** IS PREVIOUS SCAN OKAY? **
C
 2570 IF(IPSOK .EQ. 1)                                      GOTO 2580
      LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
      NIX=NX
      NX=1
      NSAMPL=1
                                                            GOTO 2130
C
C     ** MOVE MEAS DATA TO CH. A4 ARRAY (NA4)
C
 2580 READ(2'1)NA4
      CALL AMOV(MEAS(LOCA2),NA4(NSAMPL),NX)
      WRITE(2'1)NA4
      READ(2'2)NA4
      CALL AMOV(MEAS(LOCA3),NA4(NSAMPL),NX)
      WRITE(2'2)NA4
      READ(2'3)NA4
      CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      WRITE(2'3)NA4
      READ(2'4)NA4
      CALL AMOV(MEAS(LOCA5),NA4(NSAMPL),NX)
      WRITE(2'4)NA4
      READ(2'5)NA4
      CALL AMOV(MEAS(LOCA6),NA4(NSAMPL),NX)
      WRITE(2'5)NA4
      NXA4=NSAMPL+NX-1
      LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
```

```
      NIX=NX
      NX=1
      NSAMPL=1
C
C     ** JSCAN = 1 INDICATES FIRST SCAN OF 20 DETECTED **
C
      JSCAN=JSCAN+1
      IF(IPRO .EQ. 1)GOTO 2600
C
C     ******************************************************
C     ** THIS STARTS THE ACTUAL PROCESSING OF THE CHANNEL **
C     ** A4 DATA FOR CALCULATING RESPONSIVITY.            **
C
C     ******************************************************
C
C
C     **  CONVERT WAVELENGTH FOR CHANNEL A6 TO PCM COUNT          **
C     **  VALUES. THE WAVELENGTH, BRIGHTNESS OR REFLECTIVITY,     **
C     **  CHANNEL BIAS VOLTAGES, AND COEFFICIENTS FOR COMPUT-     **
C     **  ING THE VALUES ARE FOUND IN INPUT ARRAY RIN,POSI-       **
C     **  TIONS  1 THRU 28.                                       **
C
C
C
C     ** CONVERT WAVELENGTH FOR CH. 2, 3, AND 5 TO PCM VALUE **
C
      PCMA2=RIN(4)+(RIN(5)*RIN(1))
      PCMA3=RIN(11)+(RIN(12)*RIN(8))+(RIN(13)*(RIN(8)**2))+(RIN(14)*
     1       (RIN(8)**3))
      PCMA5=RIN(18)+(RIN(19)*RIN(15))+(RIN(20)*RIN(15)**2)
      GOTO 2650
C
C     **  DETERMINE IF THIS IS THE FIRST SCAN PROCESSED **
C
C
C     ** CONVERT CH. A6 **
C
 2600 PCMA6=RIN(25)+(RIN(26)*RIN(22))
C
C     ** FIND PEAK VALUE IN CHANNEL A4 (MPK4) **
C
 2650 K=NXA4+1
      READ(2'3)NA4
      MPK4=NA4(NXA4)
 2660 DO 2670 J=1,NXA4
      IF(MPK4 .GT. NA4(K-J))                               GOTO 2670
      MPK4=NA4(K-J)
 2670 CONTINUE
C
C     ** COMPENSATE FOR A4 DRIFT **
C
      IF(IPRO .EQ. 1)                                      GOTO 2800
      CA2=(PCMA2*MPK4)/983
      CA3=(PCMA3*MPK4)/983
      CA5=(PCMA5*MPK4)/983
      GOTO 2710
```

```
 2700 CALL ZERO(NA4(1),685)
      DO 2705 I=1,5
      WRITE(2'I)NA4
 2705 CONTINUE
      IPSOK=0
      NFSCN=0
      NSAMPL=1
C
C******* ANOMALY -- PEAK VALUE OF A4 NOT >95% AT END OF SCAN **
      JPAR=18
      KRET=1
      NSAM=NX
      LAN=2
      CALL SEG('DK1:ANPRO.LLL',2)
                                                       GOTO 2140
C******* FIND A4 REPRESENTATIVE VALUE FOR EACH CHANNEL **
C
 2710 DO 2720 J=3,NXA4
      IF(NA4(J) .GE. CA2)                              GOTO 2850
 2720 CONTINUE
C*******  ERROR: NO A2 VALUE FOUND IN A4 ARRAY **
      LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2730)ITIM(1),ITIM(2),SEC
 2730 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     2',I5,':',I2,':',F7.4,'  NO REPRESENTATIVE CH. A2 VALUE WAS FOUND I
     3N THE A4 ARRAY')
C
 2740 DO 2750 J=3,NXA4
      IF(NA4(J) .GE. CA3)                              GOTO 2900
 2750 CONTINUE
C******** ERROR - NO A3 VALUE FOUND IN A4 ARRAY **
C
      LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2760)ITIM(1),ITIM(2),SEC
 2760 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,'  NO REPRESENTATIVE CH. A3 VALUE FOUND IN TH
     2E A4 ARRAY')
 2770 DO 2780 J=3,NXA4
      IF(NA4(J) .GE. CA5)GOTO 2920
 2780 CONTINUE
C
C******** ERROR - NO A5 VALUE FOUND IN A4 ARRAY **
C
      LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2790)ITIM(1),ITIM(2),SEC
 2790 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,'  NO REPRESENTATIVE CH. A5 VALUE FOUND IN TH
     2E A4 ARRAY')
                                                       GOTO 2999
 2800 CA6=(PCMA6*MPK4)/983
```

```
      DO 2810 J=3,NXA4
      IF(NA4(J) .GE. CA6)                                GOTO 2970
 2810 CONTINUE
C
C********* ERROR - NO A6 VALUE FOUND IN A4 ARRAY **
C
      LLL=KCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2820)ITIM(1),ITIM(2),SEC
 2820 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,'  NO REPRESENTATIVE CH. A6 VALUE FOUND IN TH
     2E A4 ARRAY')
                                                         GOTO 2999
 2850 IF((J-5) .LE. 0)                                   GOTO 2860
      IF((J+2) .GE. NXA4)                                GOTO 2860
      JA2=J
                                                         GOTO 2740
 2860 LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DEC,ITIM,SEC)
      WRITE(6,2865)ITIM(1),ITIM(2),SEC
 2865 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,' A2 VALUE TOO NEAR START/END OF A4 ARRAY')
                                                         GOTO 2999
 2900 IF((J-5) .LE. 0)                                   GOTO 2910
      IF((J+2) .GE. NXA4)                                GOTO 2910
      JA3=J
                                                         GOTO 2770
 2910 LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2915)ITIM(1),ITIM(2),SEC
 2915 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,' A3 VALUE TOO NEAR START/END OF A4 ARRAY')
                                                         GOTO 2999
 2920 IF((J-5) .LE. 0)                                   GOTO 2930
      IF((J+2) .GE. NXA4)                                GOTO 2930
      JA5=J
      READ(2'1)NA4
      IV2=NA4(JA2-2)+NA4(JA2-1)+NA4(JA2)+NA4(JA2+1)+NA4(JA2+2)
      VJ2=IV2/5
      AVJ2=AVJ2+VJ2
      SVJ2=SVJ2+(VJ2**2)
      READ(2'2)NA4
      IV3=NA4(JA3-2)+NA4(JA3-1)+NA4(JA3)+NA4(JA3+1)+NA4(JA3+2)
      VJ3=IV3/5
      AVJ3=AVJ3+VJ3
      SVJ3=SVJ3+(VJ3**2)
      READ(2'4)NA4
      IV5=NA4(JA5-2)+NA4(JA5-1)+NA4(JA5)+NA4(JA5+1)+NA4(JA5+2)
      VJ5=IV5/5
      AVJ5=AVJ5+VJ5
      SVJ5=SVJ5+(VJ5**2)
      IF (IIN(3) .GE. 0)                                 GOTO 2925
      CALL PDUMP(VJ2,VJ2,4)
```

```
      CALL PDUMP(AVJ2,AVJ2,4)
      CALL PDUMP(SVJ2,SVJ2,4)
      CALL PDUMP(VJ3,VJ3,4)
      CALL PDUMP(AVJ3,AVJ3,4)
      CALL PDUMP(SVJ3,SVJ3,4)
      CALL PDUMP(VJ5,VJ5,4)
      CALL PDUMP(AVJ5,AVJ5,4)
      CALL PDUMP(SVJ5,SVJ5,4)
 2925 CONTINUE
      NSSCAN=NSSCAN+1
      IF(NSSCAN .GE. 20)                                      GOTO 2940
      IF(NIX .LT. IHI)                                        GOTO 2130
                                                              GOTO 2999
 2930 LLL=JCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2935)ITIM(1),ITIM(2),SEC
 2935 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,' A5 VALUE TOO NEAR START/END OF A4 ARRAY')
                                                              GOTO 2999
 2940 AVE2=AVJ2/NSSCAN
      AVE3=AVJ3/NSSCAN
      AVE5=AVJ5/NSSCAN
C
C******** FIND SQUARE ROOT OF AVERAGES SQUARED MINUS AVERAGE SQUARED  **
C
      SQAJ2=AVJ2**2
      SSDA2=(NSSCAN*SVJ2)-SQAJ2
      SQDA2=SSDA2/(NSSCAN-1)
      SDA2=SQRT(SQDA2)
C
C     ** A3 **
C
      SQAJ3=AVJ3**2
      SSDA3=(NSSCAN*SVJ3)-SQAJ3
      SQDA3=SSDA3/(NSSCAN-1)
      SDA3=SQRT(SQDA3)
C
C     ** A5 **
C
      SQAJ5=AVJ5**2
      SSDA5=(NSSCAN*SVJ5)-SQAJ5
      SQDA5=SSDA5/(NSSCAN-1)
      SDA5=SQRT(SQDA5)
C
C******** CONVERT VALUES FOR PCM COUNTS TO VOLTS **
C
      VAVE2=AVE2*.005
      VAVE3=AVE3*.005
      VAVE5=AVE5*.005
      VSD2=SDA2*.005
      VSD3=SDA3*.005
      VSD5=SDA5*.005
C
C******** SUBTRACT CHANNEL BIAS VOLTAGE ***
C
```

```
      SV2=VAVE2-RIN(3)
      SV3=VAVE3-RIN(10)
      SV5=VAVE5-RIN(17)
C
C******** COMPUTE RESPONSIVITY **
C
      RESPO2=SV2/RIN(2)
      RESPO3=SV3/RIN(9)
      RESPO5=SV5/RIN(16)
C
C******* COMPUTE NESR **
C
      VNESR2=VSD2/RESPO2
      VNESR3=VSD3/RESPO3
      VNESR5=VSD5/RESPO5
      IF (IIN(3) .GE. 0)                                      GOTO 2945
      CALL PDUMP(SDA2,SDA2,4)
      CALL PDUMP(SDA3,SDA3,4)
      CALL PDUMP(SDA5,SDA5,4)
      CALL PDUMP(SV2,SV2,4)
      CALL PDUMP(SV3,SV3,4)
      CALL PDUMP(SV5,SV5,4)
      WRITE(6,2941)RESPO2,RESPO3,RESPO5,VNESR2,VNESR3,VNESR5
 2941 FORMAT(1H ,'RESPO2 =',F15.7,'RESPO3 =',F15.7,'RESPO5 =',F15.7,
     *//,'VNESR2 =',F15.7,'VNESR3 =',F15.7,'VNESR5 =',F15.7)
 2945 CONTINUE
C
C******** STORE A2, A3 AND A5 VALUES FOR HISTORICAL FILE ***
C
      NTABUF(1)=5
      INTNUM=RESPO2
      RFRAC=RESPO2-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(4)=INTNUM
      NTABUF(5)=IFRAC
      INTNUM=VSD2
      RFRAC=VSD2-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(9)=INTNUM
      NTABUF(10)=IFRAC
      INTNUM=VNESR2
      RFRAC=VNESR2-INTNUM
      RNUM=RFRAC*10000.
      INTNUM=RNUM
      RFRAC=RNUM-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      IVAL=MEAS(3976)
      RSWL=IVAL*.005
      INTNUM=RSWL
      RFRAC=RSWL-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(19)=INTNUM
      NTABUF(20)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
```

```
2946 CALL NTRAN(12,15,LSTAT)
     IF(LSTAT+1)2947,2946,2948
2947 LUL=12
     WRITE(6,2998)LUL,LSTAT
                                                  GOTO 3000
2948 NTABUF(1)=7
     INTNUM=RESPO3
     RFRAC=RESPO3-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(4)=INTNUM
     NTABUF(5)=IFRAC
     INTNUM=VSD3
     RFRAC=VSD3-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(9)=INTNUM
     NTABUF(10)=IFRAC
     INTNUM=VNESR3
     RFRAC=VNESR3-INTNUM
     RNUM=RFRAC*10000.
     INTNUM=RNUM
     RFRAC=RNUM-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(14)=INTNUM
     NTABUF(15)=IFRAC
     CALL NTRAN(12,1,50,NTABUF(1))
2949 CALL NTRAN(12,15,LSTAT)
     IF(LSTAT+1)2950,2949,2951
2950 LUL=12
     WRITE(6,2998)LUL,LSTAT
                                                  GOTO 3000
2951 NTABUF(1)=9
     INTNUM=RESPO5
     RFRAC=RESPO5-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(4)=INTNUM
     NTABUF(5)=IFRAC
     INTNUM=VSD5
     RFRAC=VSD5-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(9)=INTNUM
     NTABUF(10)=IFRAC
     INTNUM=VNESR5
     RFRAC=VNESR5-INTNUM
     RNUM=RFRAC*10000.
     INTNUM=RNUM
     RFRAC=RNUM-INTNUM
     IFRAC=RFRAC*10000.
     NTABUF(14)=INTNUM
     NTABUF(15)=IFRAC
     CALL NTRAN(12,1,50,NTABUF(1))
2952 CALL NTRAN(12,15,LSTAT)
     IF(LSTAT+1)2953,2952,2954
2953 LUL=12
     WRITE(6,2998)LUL,LSTAT
                                                  GOTO 3000
2954 IDONE=IDONE+3
```

```
      IRSFLG=IRSFLG+1
      JCMPLT(2)=1
      AVJ2=0.
      AVJ3=0.
      AVJ5=0.
      VSD2=0.
      VSD3=0.
      VSD5=0.
      SVJ3=0.
      SVJ2=0.
      SVJ5=0.
      JSCAN=0
      NSSCAN=0
      NFSCN=0
      NX=0
      NSAMPL=0
      NIX=0
      IPSOK=0
                                                      GOTO 2999
C     *****************************************************************
C     * PUT ZERO VALUES IN HISTORICAL FILE FOR BAD RESPONSIVITY CAL    *
C     *****************************************************************
 2955 IF(IDONE .GE. 11)                               GOTO 4000
 2960 NTABUF(1)=5
      CALL ZERO(NTABUF(2),49)
      CALL NTRAN(12,1,50,NTABUF(1))
 2961 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)2962,2961,2963
 2962 LUL=12
      WRITE(6,2998)LUL,LSTAT
                                                      GOTO 3000
 2963 NTABUF(1)=7
      CALL NTRAN(12,1,50,NTABUF(1))
 2964 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)2965,2964,2966
 2965 LUL=12
      WRITE(6,2998)LUL,LSTAT
                                                      GOTO 3000
 2966 NTABUF(1)=9
      CALL NTRAN(12,1,50,NTABUF(1))
 2967 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)2968,2967,2969
 2968 LUL=12
      WRITE(6,2998)LUL,LSTAT
                                                      GOTO 3000
 2969 IDONE=IDONE+3
      IRSFLG=IRSFLG+1
      NX=0
      NSAMPL=0
      NFSCN=0
      NIX=0
      NSSCAN=0
      IPSOK=0
      JSCN6=0
                                                      GOTO 2999
C
```

```
C******** COMPUTE CH. A6 RESPONSIVITY *****
C
 2970 IF((J-5) .LE. 0)                                  GOTO 2980
      IF((J+2) .GE. NXA4)                               GOTO 2980
      READ(2'5)NA4
      IV6=NA4(J-2)+NA4(J-1)+NA4(J)+NA4(J+1)+NA4(J+2)
      VJ6=IV6/5
      AVJ6=AVJ6+VJ6
      SVJ6=SVJ6+(VJ6**2)
      ISCN6=ISCN6+1
      IF (IIN(3) .GE. 0)                                GOTO 2971
      CALL PDUMP(VJ6,VJ6,4)
      CALL PDUMP(AVJ6,AVJ6,4)
      CALL PDUMP(SVJ6,SVJ6,4)
 2971 CONTINUE
C
C******* DO WE HAVE TWENTY SCANS OF A6 DATA ? **
C
      IF(ISCN6 .GE. 20)                                 GOTO 2975
      IF(NIX .LT. IHI)                                  GOTO 2130
      GOTO 2999
C
C******** YES - COMPUTE AVERAGE A6 VALUE **
C
 2975 AVE6=AVJ6/ISCN6
      SQAJ6=AVJ6**2
      SSDA6=(ISCN6*SVJ6)-SQAJ6
      SQDA6=SSDA6/(ISCN6-1)
      SDA6=SQRT(SQDA6)
                                                        GOTO 2990
C     ****************************************************************
C     *  PUT ZERO VALUES IN HISTORICAL FILE FOR BAD HEATED CAL       *
C     ****************************************************************
 4000 IF(IDONE .EQ. 12)                                 GOTO 2999
 2976 NTABUF(1)=11
      CALL ZERO(NTABUF(2),49)
      CALL NTRAN(12,1,50,NTABUF(1))
 2977 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)2978,2977,2979
 2978 LUL=12
      WRITE(6,2998)LUL,LSTAT
                                                        GOTO 3000
 2979 IDONE=IDONE+1
      IRSFLG=IRSFLG+1
      NX=0
      NSAMPL=0
      NFSCN=0
      NIX=0
      NSSCAN=0
      IPSOK=0
      JSCN6=0
                                                        GOTO 2999
 2980 LLL=KCMPLT(1)
      DSEC=SSTIME(1,LLL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,2985)ITIM(1),ITIM(2),SEC
```

```
 2985 FORMAT(1H ,'* * * * * ERROR - DURING RESPONSIVITY ON CAL PERIOD AT
     1',I5,':',I2,':',F7.4,' A6 VALUE TOO NEAR START/END OF A4 ARRAY')
                                                        GOTO 2999
C
C******** CONVERT PCM VALUES TO VOLTS **
C
 2990 VAVE6=AVE6*.005
      VSD6=SDA6*.005
C
C******** SUBTRACT CHANNEL BIAS VOLTAGE **
C
      SV6=VAVE6-RIN(24)
C
C******** COMPUTE TEMPERATURE FOR HEATED, REFERENCE AND DICHROIC TEMPS.*
C
      IHT=FLOAT(MEAS(376))
      IRT=FLOAT(MEAS(484))
      IDT=FLOAT(MEAS(322))
      JJNT=1
      NDICAL=MEAS(322)
      NTRCAL=MEAS(484)
      THEAT=CNVTMP(1)+(CNVTMP(2)*IHT)+(CNVTMP(3)*(IHT**2))+(CNVTMP(4)*
     1(IHT**3))+(CNVTMP(5)*(IHT**4))+(CNVTMP(6)*(IHT**5))
C
      TREF=CNVTMP(37)+(CNVTMP(38)*IRT)+(CNVTMP(39)*(IRT**2))+(CNVTMP(40)
     1*(IRT**3))+(CNVTMP(41)*(IRT**4))+(CNVTMP(42)*(IRT**5))
C
      TDICH=CNVTMP(49)+(CNVTMP(50)*IDT)+(CNVTMP(51)*(IDT**2))+
     1(CNVTMP(52)*(IDT**3))+(CNVTMP(53)*(IDT**4))+(CNVTMP(54)*(IDT**5))
C
      XPNH=RIN(22)*(THEAT+273)
      XPNR=RIN(22)*(TREF+273)
      XPND=RIN(22)*(TDICH+273)
      XH=14388/XPNH
      XR=14388/XPNR
      XD=14388/XPND
      WVL=RIN(22)**5
      BWH=11909/(WVL*((EP**XH)-1))
      BWR=11909/(WVL*((EP**XR)-1))
      BWD=11909/(WVL*((EP**XD)-1))
      ZIH=(RIN(23)*BWH)+((1-RIN(23))*BWD)
      RESPO6=SV6/(ZIH-BWR)
      VNESR6=VSD6/RESPO6
      IF (IIN(3) .GE. 0)                                GOTO 2996
      CALL PDUMP(SDA6,SDA6,4)
      CALL PDUMP(SV6,SV6,4)
      CALL PDUMP(THEAT,THEAT,4)
      CALL PDUMP(TREF,TREF,4)
      CALL PDUMP(TDICH,TDICH,4)
      WRITE(6,2995)RESPO6,VNESR6
 2995 FORMAT(1H ,'RESPO6 =',F15.7,' VNESR6 =',F15.7)
 2996 CONTINUE
C
C******** STORE A6 VALUES FOR HISTORICAL FILE **
C
      NTABUF(1)=11
```

```
      INTNUM=RESPO6
      RFRAC=RESPO6-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(4)=INTNUM
      NTABUF(5)=IFRAC
      INTNUM=VSD6
      RFRAC=VSD6-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(9)=INTNUM
      NTABUF(10)=IFRAC
      INTNUM=VNESR6
      RFRAC=VNESR6-INTNUM
      RNUM=RFRAC*10000.
      INTNUM=RNUM
      RFRAC=RNUM-INTNUM
      IFRAC=RFRAC*10000.
      NTABUF(14)=INTNUM
      NTABUF(15)=IFRAC
      CALL NTRAN(12,1,50,NTABUF(1))
 2991 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)2992,2991,2993
 2992 LUL=12
      WRITE(6,2998)LUL,LSTAT
                                                    GOTO 3000
 2993 IDONE=IDONE+1
      IRSFLG=IRSFLG+1
      KCMPLT(2)=1
      AVJ6=0.
      SVJ6=0.
      VSD6=0.
      JSCN6=0
      NSSCAN=0
      NFSCN=0
      NX=0
      NSAMPL=0
      NIX=0
      IPSOK=0
 2998 FORMAT(1H0,'NTRAN WRITE ERROR ON UNIT',I4,' STATUS WORD',I4)
 2999 CALL SEG('DK1:QAEXEC.GGG',0,0)
 3000 CALL SEGRET
      END

ROUTINES CALLED:
SEG    , AMOV   , ZERO   , STOHMS, PDUMP  , SQRT   , NTRAN
FLOAT  , SEGRET

SWITCHES = /ON,/SU,/CO

BLOCK       LENGTH
MAIN.    6390    (030754)*
PRNDX    54      (000154)
ANDAT    91      (000266)
DCDATA   3993    (017462)
DCARGN   28      (000070)
RESDAT   712     (002620)
SIXSV    1       (000002)
CALRT    3       (000006)
HISDAT   8       (000020)
INPUT    158     (000474)
INTNDX   5       (000012)
RECPTR   1       (000002)
TIMES    522     (002024)
```

```
**COMPILER ----- CORE**
   PHASE       USED  FREE
DECLARATIVES 04762 12504
EXECUTABLES  06071 11195
ASSEMBLY     04055 16128
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:TMLOOP.OBJ,LP:<DK1:TMLOOP.FTN/ON/SU/CO:99
```

```
C
C
C      LOAD MODULE: TMLOOP
C
C
C      **** TYPE STATEMENT ****
C
       DOUBLE PRECISION SSTIME,FRSPRC,LASPRC,LASNXT
       DOUBLE PRECISION STOP, START, TOL
       DOUBLE PRECISION  DELTME,   DEL
       DOUBLE PRECISION  FRTEST
C
C
       INTEGER FMT, FRSTR, FRSTP
       INTEGER  ANSTAT
       INTEGER PRNDX1,PRNDX2,PRNDX3
       DIMENSION PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
C
C      ***** COMMON STATEMENTS *****
C
       COMMON /DCARGN/ START,  STOP,     TOL,      IFLAGG,
     1                 FMT,    IDKRTP,   FRSTR(2), FRSTP(2),
     2                 IDBLE,  LU,       ISIZE,    NTH,
     3                 MAX,    NAV,      INIT,     IDDC,
     4                 ISTAT
       COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
       COMMON /DCDATA/ MEAS(3993)
       COMMON /INPUT/  IIN(102),          RIN(28)
       COMMON /PRNDX/ PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
       COMMON /TIMES/ BJD,SSTIME(2,65)
C
C      **** EQUIVALENCE STATEMENTS *****
C
C
       EQUIVALENCE (IIN(16),NOVRL),(IIN(17),NCALP),(IIN(18),NWCAL)
C
C
C      * DETERMINE TIME GROUP (I) AND ARRAY INDEX (L) *
       CALL AMOV(MEAS(1),FRSPRC,4)
       CALL AMOV(MEAS(85),DELTME,4)
       CALL AMOV(MEAS(141),DEL,4)
       CALL ZERO(PRNDX1(1,1),18)
       CALL ZERO(PRNDX2(1,1),18)
       CALL ZERO(PRNDX3(1,1),18)
       LASPRC=FRSPRC + ((MAX-1) * DELTME)
       DO 200 I=1,9
       GOTO(20,30,40,50,60,70,80,90,100) I
    20 KONT=NOVRL
       J=NCALP
       GOTO 110
C
C
C
    30 KONT=NOVRL+NCALP
       J=NWCAL
```

```
      GOTO 110
C
C
C
   40 KONT=NOVRL+NCALP+NWCAL
      J=NCALP
      GOTO 110
C
C
C
   50 KONT=NOVRL+(2*NCALP)+NWCAL
      J=NCALP
      GOTO 110
C
C
C
   60 KONT=NOVRL+(3*NCALP)+NWCAL
      J=NCALP
      GOTO 110
C
C
C
   70 KONT=NOVRL+(4*NCALP)+NWCAL
      J=NCALP
      GOTO 110
C
C
C
   80 KONT=NOVRL+(5*NCALP)+NWCAL
      J=NCALP
      GOTO 110
C
C
C
   90 KONT=NOVRL+(6*NCALP)+NWCAL
      J=NCALP
      GOTO 110
C
C
  100 KONT=NOVRL+(7*NCALP)+NWCAL
      J=NWCAL
C
C
  110 DO 150 M=1,J
C
      IL=KONT+M
C
C
      IF(LASPRC .LT. SSTIME(1,IL))GOTO 150
C
      IF(FRSPRC .GT. SSTIME(2,IL))GO TO 150
      GOTO 220
  150 CONTINUE
  200 CONTINUE
C
  210                                                   GOTO 9999
```

```
C
  220 GOTO(970,230,972,230,230,230,976,977,978) I
  230 FRTEST=FRSPRC
      NLOOP=MAX * 32
      DO 240 NL=1,NLOOP
      IF(FRTEST .GE. SSTIME(1,IL))                        GOTO 250
      FRTEST=FRTEST + DEL
  240 CONTINUE
                                                          GOTO 260
  250 PRNDX1(1,I)=NL/32 + 1
      PRNDX2(1,I)=NL/16 + 1
      PRNDX3(1,I)=NL
  260 LASNXT=LASPRC+DELTME
      IF(LASNXT .LE. SSTIME(2,IL))                        GOTO 960
      NCURR = NLOOP + 1
      DO 950 KL=1,NLOOP
      LASNXT = LASNXT - DEL
      NCURR = NCURR - 1
      IF(LASNXT .GT. SSTIME(2,IL))                        GOTO 950
      PRNDX1(2,I) = NCURR/32 + 1
      PRNDX2(2,I) = NCURR/16 + 1
      PRNDX3(2,I) = NCURR
                                                          GOTO 200
  950 CONTINUE
  960 PRNDX1(2,I) = 18
      PRNDX2(2,I) = 36
      PRNDX3(2,I) = 576
                                                          GOTO 200
  970 L=IL
                                                          GOTO 230
  972 LBIAS=IL
                                                          GOTO 230
  976 LHEAT=IL
                                                          GOTO 230
  977 LSWL=IL
                                                          GOTO 230
  978 LWLW=IL
                                                          GOTO 230
 9999 CONTINUE
      IF (IIN(3) .GE. 0)                                  GOTO 1000
      CALL PDUMP(PRNDX1(1,1),PRNDX1(2,9),1,PRNDX2(1,1),PRNDX2(2,9),1,
     *           PRNDX3(1,1),PRNDX3(2,9),1)
 1000 CONTINUE
      CALL SEG('DK1:QAEXEC,GGG',0,0)
      END

ROUTINES CALLED:
AMOV   , ZERO   , PDUMP  , SEG

SWITCHES = /ON,/SU,/CO

BLOCK        LENGTH
MAIN.    737     (002702)*
DCARGN   28      (000070)
INTNDX   5       (000012)
DCDATA   3993    (017462)
INPUT    158     (000474)
PRNDX    54      (000154)
TIMES    522     (002024)
```

FORTRAN V004A 17:45:40 27-JUL-76 PAGE 4

```
**COMPILER ----- CORE**
   PHASE      USED  FREE
DECLARATIVES 03750 13516
EXECUTABLES  04499 12767
ASSEMBLY     01613 18570
```

```
#
$EOD
$RUN FORTRN
FORTRAN  V004A
#DK1:WVLPRO.OBJ,LP:<DK1:WVLPRO.FTN/ON/SU/CO:99
```

```
C***********************************************************************
C*                                                                     *
C*        LOAD MODULE: WVLPRO                                          *
C*                                                                     *
C                                                                      *
      DOUBLE PRECISION RTME,SSTIME,DSEC
      DOUBLE PRECISION ASIM(3,5),A0,A1,A2,XA4,YA4,SUMXA4,SQXA4
      DOUBLE PRECISION ADET(4)
      DOUBLE PRECISION  SNUM,     SNA4,     SUMX2,    SUMX3,    SUMX4
      DOUBLE PRECISION  SUMXY,    SUMX2Y,   SNCHAN,   XNA4
      INTEGER  ANSTAT
      INTEGER ANCNTR,ANREC
      INTEGER PRNDX1,PRNDX2,PRNDX3
      DIMENSION ADET(4), NTABUF(50)
      DIMENSION JCHN(3)
      DIMENSION NCHAN(685), ASIM(3,5),V(2)
      DIMENSION JCMPLT(2),KCMPLT(2),NA4(685),ITIM(2)
      COMMON /PRNDX/ PRNDX1(2,9),PRNDX2(2,9),PRNDX3(2,9)
      COMMON /ANDAT/ JPAR,NSAM,LAN,MACT,ANSTAT(15),KRET,
     *               ANCNTR(15),ANREC(15),TANCTR(18),IRESFG,
     *               IWLFG, IRAMFG,LZNE,IPG
      COMMON / A4CHNL /  IA4(685)
      COMMON / DCDATA /  MEAS(3993)
      COMMON /WVLDAT/  JCMPLT(2),  LMTLO,  LMTHI
      COMMON /RESDAT/ MCMPLT(2),KCMPLT(2),NA4(685),NFSCN,NX,NSAMPL,NIX
     *,AVJ2, AVJ3, AVJ5, AVJ6,SQYA4,SUMYA4,JSCN6,IWLSCN
     *,   IPSOK
      COMMON /HISDAT/   IRMFLG,   IRSFLG,   IBVFLG,   IWLFLG,
     *                  ICLHDR,   IWLHDR,   IDONE,    IBVHDR
      COMMON /INPUT/ IIN(102),RIN(28)
      COMMON / INTNDX / L,  LBIAS,  LHEAT,  LSWL,  LWLW
      COMMON  /SIXSV/  ISIX
      COMMON /TIMES/ BJD, SSTIME(2,65)
C
C
      EQUIVALENCE (IIN(22),ISYNC),(IIN(23),IEND),(IIN(24),ISMIN),
     1            (IIN(25),ISMAX)
      EQUIVALENCE (NCHAN,IA4),  (IIN(61),IWLLO),   (IIN(62),IWLHI)
C
      DATA JCHN /  3,  5,  2/
C
C     ** START SCAN START/STOP DETECTION **
C     ** NFSCN = 1 SAYS NOT LOOKING FOR VERY FIRST SCAN **
C
      IF (IIN(3) .GE. 0)                                  GOTO 2001
      WRITE(6,2000)
 2000 FORMAT(1H ,'ENTERING WVLPRO')
 2001 IF(PRNDX3(1,9) .EQ. 0 .AND. IWLHDR .EQ. 1)          GOTO 3836
      IF(JCMPLT(1) .EQ. 0)                                GOTO 2010
      IF(JCMPLT(1) .EQ. LWLW)                             GOTO 2020
      IF(JCMPLT(2) .EQ. 1)                                GOTO 2010
      JPAR=19
      LAN=1
      KRET=3
      NSAM=0
```

```
      CALL SEG('DK1:ANPRO.LLL',2)
                                                          GOTO 3836
 2010 JCMPLT(1)=LWLW
      JCMPLT(2)=0
                                                          GOTO 2030
 2020 IF(JCMPLT(2) .EQ. 1)                                GOTO 3999
 2030 ILO=PRNDX3(1,9)
      IHI=PRNDX3(2,9)
      IF(ICHAN .GT. 1)                                    GOTO 2100
      ICHAN=1
 2100 IF(NFSCN .EQ. 1)GOTO 2110
      NSAMPL=1
C
C     ** SET UP MEAS ARRAY LOCATORS FOR ALL SIX CHANNELS **
C
 2110 ILL=ILO-1
      LOCA1=520+ILL
      LOCA2=1096+ILL
      LOCA3=1672+ILL
      LOCA4=2248+ILL
      LOCA5=2824+ILL
      LOCA6=3400+ILL
C
C     ** NX IS THE SCAN SAMPLE COUNTER **
C
 2120 NX=0
      IF(ISTA .EQ. 1)                                     GOTO 2200
C
C     ** CHECK CHANNEL A6 FOR SYNC PULSE (ISYNC) **
C
 2130 IF(MEAS(LOCA6+NX) .LT. ISYNC)GOTO 2160
      ISIX=ISIX+1
      IF(ISIX .GE. 2)                                     GOTO 2160
      IF(NX-1 .LT. 0)GOTO 2170
C
C     ** CHECK CHANNEL A4 FOR MINIMUM PEAK COUNT (IEND) **
C
      IF(MEAS(LOCA4+(NX-1)) .LT. IEND)                    GOTO 2700
C
C     ** CHECK SYNC PULSES ON OTHER CHANNELS **
C
 2140 IF(MEAS(LOCA1+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA2+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA3+NX) .LT. ISYNC)GOTO 2190
      IF(MEAS(LOCA5+NX) .LT. ISYNC)GOTO 2190
      GOTO 2500
C     ** CHECK FOR SCAN TOO LONG (> ISMAX) **
C
 2160 ISIX=0
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2300
      ITST=ILO+NX+NIX-1
      IF(ITST .GE. IHI)                                   GOTO 2180
 2170 NX=NX+1
      GOTO 2130
C
C
```

```
 2180 IF(NFSCN .EQ.0)GOTO 3999
C
C     ** MOVE PART OF SCAN TO CH. A4 ARRAY (NA4)
C
      CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      IBACK=1
      GOTO(3850,3900,3950)ICHAN
 2185 NSAMPL=NSAMPL+NX
      NFSCN=1
      NIX=0
                                                    GOTO 3999
C
C     ** CHECK FOR END OF MEAS ARRAY OR DATA PERIOD **
C
 2190 IF((NX+1) .GT. IHI)GOTO 2400
      II=NX+1
      GOTO 2210
 2200 II=NX
C
C     ** CHECK NEXT VALUES IN OTHER CHANNELS FOR SYNC PULSES **
C
 2210 IF(MEAS(LOCA1+II) .LT. ISYNC)GOTO 2221
      IF(MEAS(LOCA2+II) .LT. ISYNC)GOTO 2222
      IF(MEAS(LOCA3+II) .LT. ISYNC)GOTO 2223
      IF(MEAS(LOCA5+II) .LT. ISYNC)GOTO 2225
      GOTO 2500
C
C     ** ANOMALY - BAD SYNC PULSE -- SCAN NOT REJECTED **
C
 2221 LAN=1
      GOTO 2230
 2222 LAN=2
      GOTO 2230
 2223 LAN=3
      GOTO 2230
 2225 LAN=5
 2230 JPAR=16
      NSAM=NX
      KRET=3
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2500
C
C     ** PREVIOUS SCAN INVALID -- ZERO NA4 ARRAY **
C
 2300 CALL ZERO(NA4(1),685)
      CALL ZERO(NCHAN(1),685)
      NSAMPL=1
C     ** ANOMALY -- BAD SYNC PULSE ON CH. A6 **
      NFSCN=1
      JPAR=16
      NSAM=NX
      LAN=6
      KRET=3
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2170
C
```

```
C       ** END OF MEAS ARRAY -- MOVE GOOD DATA TO NA4 ARRAY **
C
 2400 ISTA=1
      CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      IBACK=2
      GOTO(3850,3900,3950)ICHAN
 2450 CONTINUE
                                                            GOTO 3999
C
C       ** CHECK FOR CORRECT NUMBER OF SCANS **
C
 2500 IF(NFSCN .EQ. 0)                                      GOTO 2530
      IF((NSAMPL+NX) .GT. ISMAX)GOTO 2510
      IF((NSAMPL+NX) .LT. ISMIN)GOTO 2510
      IPSOK=1
      GOTO 2530
C
C       ** ANOMALY -- SCAN TOO LONG OR TOO SHORT **
C
 2510 JPAR=18
      NSAM=NX
      IPSOK=0
      NFSCN=0
      LAN=NSAMPL+NX
      KRET=3
      CALL SEG('DK1:ANPRO.LLL',2)
      IHI=IHI-NIX
C
C       ** CHECK FOR FIRST CH. A4 SAMPLE LESS THAN 50 PCM CNTS **
C
 2530 II=LOCA4+NX+1
      IF(MEAS(II) .LT. 50)GOTO 2560
      NFSCN=0
C
C       ** ANOMALY -- FIRST SAMPLE IN A4 TOO HIGH **
C
      JPAR=18
      NSAM=NX
      LAN=2
      KRET=3
      CALL SEG('DK1:ANPRO.LLL',2)
      GOTO 2570
 2560 NFSCN=1
C
C       ** IS PREVIOUS SCAN OKAY? **
C
 2570 IF(IPSOK .EQ. 1)                                      GOTO 2580
      LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
      NIX=NX
      NX=1
      NSAMPL=1
```

```
                                                         GOTO 2130
C
C     ** MOVE MEAS DATA TO CH. A4 ARRAY (NA4)
C
 2580 CALL AMOV(MEAS(LOCA4),NA4(NSAMPL),NX)
      IBACK=3
      GOTO(3850,3900,3950)ICHAN
 2600 NXA4=NSAMPL+NX-1
      LOCA1=LOCA1+NX
      LOCA2=LOCA2+NX
      LOCA3=LOCA3+NX
      LOCA4=LOCA4+NX
      LOCA5=LOCA5+NX
      LOCA6=LOCA6+NX
      NIX=NX
      NX=1
      NSAMPL=1
C
C     ** JSCAN = 1 INDICATES FIRST SCAN OF 20 DETECTED **
C
      JSCAN=JSCAN+1
                                                         GOTO 3610
 2700 CALL ZERO(NA4(1),685)
      CALL ZERO(NCHAN(1),685)
      IPSOK=0
      NFSCN=0
      NSAMPL=1
      JPAR=18
      KRET=3
      NSAM=NX
      LAN=2
      CALL SEG('DK1:ANPRO.LLL',2)
      IHI=IHI-NIX
                                                         GOTO 2140
C
C*************************************************************************
C*                                                                       *
C*        DETERMINE WHICH CHANNEL IS BEING USED (ICHAN = )               *
C*                                                                       *
C*************************************************************************
C
C******* FIND LOW AND HIGH LIMITS IN A4 ARRAY **
C
 3610 DO 3620 J=3,NXA4
      IF(NA4(J) .GE. IWLLO)                              GOTO 3640
 3620 CONTINUE
 3630 LL=JCMPLT(1)
      DSEC=SSTIME(1,LL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,3635)ITIM(1),ITIM(2),SEC
 3635 FORMAT(1H ,'* * * * * ERROR - W/L CAL PERIOD AT',I5,':',I2,':',F7.
     14,' NO A4 LOWER LIMIT FOUND * * * * *')
      IF(NIX .LT. IHI)                                   GOTO 2130
                                                         GOTO 3999
C
C******** HIGH LIMITS **
```

```
C
 3640 LMTLO=J
      IF (IIN(3) .GE. 0)                                       GOTO 3641
      WRITE(6,4000)LMTLO
 3641 CONTINUE
      DO 3650 J=3,NXA4
      IF(NA4(J) .GE. IWLHI)                                    GOTO 3670
 3650 CONTINUE
 3660 LL=JCMPLT(1)
      DSEC=SSTIME(1,LL)
      CALL STOHMS(DSEC,ITIM,SEC)
      WRITE(6,3665)ITIM(1),ITIM(2),SEC
 3665 FORMAT(1H ,'* * * * * ERROR - W/L CAL PERIOD AT',I5,':',I2,':',F7.
     14,' NO A4 UPPER LIMIT FOUND * * * * *')
      IF(NIX .LT. IHI)                                         GOTO 2130
                                                               GOTO 3999
C******** FIND PEAK VALUE IN A4 ARRAY **
C
 3670 LMTHI=J
      IF (IIN(3) .GE. 0)                                       GOTO 3671
      WRITE(6,4001)LMTHI
 3671 CONTINUE
      IPK4=NA4(NXA4)
      DO 3680 I=3,NXA4
      IF(NA4(I) .LE. IPK4)                                     GOTO 3680
      IPK4=NA4(I)
 3680 CONTINUE
      IF (IIN(3) .GE. 0)                                       GOTO 3681
      WRITE(6,4002)IPK4
 3681 CONTINUE
C
C******** TEST SELECTED CHANNEL ARRAY FOR SATURATED DATA **
 3700 ITEST=0
      DO 3720 K=LMTLO,LMTHI
      IF(ITEST .EQ. 0)                                         GOTO 3705
      ISAT=999
      GOTO 3710
 3705 ISAT=949
 3710 ITEST=ITEST+1
      IF(NCHAN(K) .GT. ISAT)                                   GOTO 3725
 3720 CONTINUE
                                                               GOTO 3740
 3725 ICHAN=ICHAN+1
      IF(ICHAN .GT. 3)                                         GOTO 3730
      ITEST=0
      NFSCN=0
                                                               GOTO 2100
 3730 JPAR=19
      NSAM=JCMPLT(1)
      LAN=0
      KRET=3
      CALL SEG('DK1:ANPRO.LLL',2)
      JCMPLT(2)=1
                                                               GOTO 3999
C
C******** FIND UPPER AND LOWER LIMITS IN SELECTED CHANNEL **
```

```
      C
       3740 IRAN=NCHAN(LMTLO)
            IF (IIN(3) .GE. 0)                              GOTO 3741
            WRITE(6,4003)IRAN
       3741 CONTINUE
            DO 3745 K=LMTLO,LMTHI
            IF(NCHAN(K) .LE. IRAN)                          GOTO 3745
            JLO=K
            IRAN=NCHAN(K)
       3745 CONTINUE
            IF (IIN(3) .GE. 0)                              GOTO 3755
            WRITE(6,3746)JLO,IRAN
       3746 FORMAT(1H ,'JLO =',I5,' IRAN =',I5)
       3755 ITOP=LMTHI-LMTLO+1
            IF (IIN(3) .GE. 0)                              GOTO 3756
            WRITE(6,4004)ITOP
       3756 CONTINUE
            I=LMTHI+1
            DO 3760 K=1,ITOP
            IF(NCHAN(I-K) .EQ. IRAN)                        GOTO 3765
       3760 CONTINUE
                                                            GOTO 3770
       3765 JHI=I-K
            IF (IIN(3) .GE. 0)                              GOTO 3770
            WRITE(6,4005)JHI
       3770 INEW=IRAN/2
            I=JLO
            ISTOP=JLO-LMTLO
            DO 3775 K=1,ISTOP
            IF(NCHAN(I-K) .LE. INEW)                        GOTO 3780
       3775 CONTINUE
            JPAR=18
            NSAM=LMTLO
            LAN=5
            KRET=3
            CALL SEG('DK1:ANPRO.LLL',2)
            IF(NIX .LT. IHI)                                GOTO 2130
                                                            GOTO 3999
       3780 KLO=I-K
            IF (IIN(3) .GE. 0)                              GOTO 3781
            WRITE(6,4006)KLO
       3781 CONTINUE
            DO 3785 K=JLO,LMTHI
            IF(NCHAN(K) .LE. INEW)                          GOTO 3790
       3785 CONTINUE
            JPAR=18
            NSAM=LMTLO
            LAN=6
            KRET=3
            CALL SEG('DK1:ANPRO.LLL',2)
            IF(NIX .LT. IHI)                                GOTO 2130
                                                            GOTO 3999
       3790 KHI=K
            IF (IIN(3) .GE. 0)                              GOTO 3792
            WRITE(6,3791)KHI
       3791 FORMAT(1H ,'KHI =',I5)
```

```
3792 CONTINUE
     IF(KLO .GT. LMTLO)                                    GOTO 379
     JPAR=18
     NSAM=LMTLO
     LAN=4
     KRET=3
     CALL SEG('DK1:ANPRO.LLL',2)
     IF(NIX .LT. IHI)                                      GOTO 2130
                                                           GOTO 3999
3795 IF(KHI .LT. LMTHI)                                    GOTO 3800
     JPAR=18
     NSAM=LMTLO
     LAN=3
     KRET=3
     CALL SEG('DK1:ANPRO.LLL',2)
     IF(NIX .LT. IHI)                                      GOTO 2130
                                                           GOTO 3999
3800 INUM=KHI-KLO+1
     IF (IIN(3) .GE. 0)                                    GOTO 3801
     WRITE(6,4007)INUM
3801 CONTINUE
     IF(INUM .LT. 6)                                       GOTO 3805
     IF(INUM .GT. 10)                                      GOTO 3810
                                                           GOTO 3815
3805 LAN=8
                                                           GOTO 381
3810 LAN=7
3811 JPAR=18
     NSAM=LMTLO
     KRET=3
     CALL SEG('DK1:ANPRO.LLL',2)
     IF(NIX .LT. IHI)                                      GOTO 2130
                                                           GOTO 3999
3815 SNCHAN=0.
     SNA4=0.
     SUMXY=0.
     SUMX2Y=0.
     SUMX2=0.
     SUMX3=0.
     SUMX4=0.
     DO 3816 L=KLO, KHI
     SNCHAN=SNCHAN+NCHAN(L)
     XNA4=NA4(L)
     SNA4=SNA4+XNA4
     SUMXY=SUMXY+(NCHAN(L)*XNA4)
     SUMX2Y=SUMX2Y+((XNA4**2)*NCHAN(L))
     SUMX2=SUMX2+(XNA4**2)
     SUMX3=SUMX3+(XNA4**3)
     SUMX4=SUMX4+(XNA4**4)
     SNUM=INUM
     IF (IIN(3) .GE. 0)                                    GOTO 3816
     CALL PDUMP(SUMX2Y,SUMX2Y,4)
     CALL PDUMP(XNA4,XNA4,4)
     CALL PDUMP(SNA4,SNA4,4)
     CALL PDUMP(SUMXY,SUMXY,4)
     CALL PDUMP(SUMX2Y,SUMX2Y,4)
```

```
      CALL PDUMP(SUMX2,SUMX2,4)
      CALL PDUMP(SUMX3,SUMX3,4)
      CALL PDUMP(SUMX4,SUMX4,4)
      CALL PDUMP(SNUM,SNUM,4)
 3816 CONTINUE
C
C******** SET UP ARRAY 'A' FOR SIMEQ SUBROUTINE **
C
      ASIM(1,1)=SNUM
      ASIM(1,2)=SNA4
      ASIM(1,3)=SUMX2
      ASIM(1,4)=SNCHAN
      ASIM(2,1)=SNA4
      ASIM(2,2)=SUMX2
      ASIM(2,3)=SUMX3
      ASIM(2,4)=SUMXY
      ASIM(3,1)=SUMX2
      ASIM(3,2)=SUMX3
      ASIM(3,3)=SUMX4
      ASIM(3,4)=SUMX2Y
      IF (IIN(3) .GE. 0)                                GOTO 3817
      CALL PDUMP(ASIM(1,1),ASIM(3,5),5)
 3817 CONTINUE
C
C
C
C     *****************************************************************
C     *    SOLVE THE SIMULTANEOUS EQUATIONS FOR THE PARABOLIC BEST FIT *
C     *****************************************************************
      DO 3818 IDET=1,4
      J = 1
      L = 2
      N = 3
      IF (IDET .EQ. 2) J=4
      IF (IDET .EQ. 3) L=4
      IF (IDET .EQ. 4) N=4
      ADET(IDET) = (ASIM(1,J)*ASIM(2,L)*ASIM(3,N)) +
     *             (ASIM(2,J)*ASIM(3,L)*ASIM(1,N)) +
     *             (ASIM(3,J)*ASIM(1,L)*ASIM(2,N)) -
     *             (ASIM(1,J)*ASIM(3,L)*ASIM(2,N)) -
     *             (ASIM(2,J)*ASIM(1,L)*ASIM(3,N)) -
     *             (ASIM(3,J)*ASIM(2,L)*ASIM(1,N))
 3818 CONTINUE
      IF (IIN(3) .GE. 0)                                GOTO 3819
      CALL PDUMP(ADET(1),ADET(4),5)
 3819 CONTINUE
C     *****************************************************************
C     *    CALCULATE A0                                               *
C     *****************************************************************
      A0 = ADET(2) / ADET(1)
C     *****************************************************************
C     *    CALCULATE A1                                               *
C     *****************************************************************
      A1 = ADET(3) / ADET(1)
C     *****************************************************************
C     *    CALCULATE A2                                               *
```

```
C     *****************************************************************
      A2 = ADET(4) / ADET(1)
      IF (IIN(3) .GE. 0)                                    GOTO 3824
      CALL PDUMP(A0,A0,5)
      CALL PDUMP(A1,A1,5)
      CALL PDUMP(A2,A2,5)
      DO 3823 JEST=KLO,KHI
      XA4 = NA4(JEST)
      YA4 = A0 + (A1 * XA4) + (A2 * (XA4**2))
      WRITE (6,4010) NA4(JEST),NCHAN(JEST),YA4
 3823 CONTINUE
C
C
C
 3824 XA4=-1*(A1/(2*A2))
      XA4=(XA4*983.)/IPK4
      YA4=A0+(A1*XA4)+(A2*(XA4**2))
      IF (IIN(3) .GE. 0)                                    GOTO 3826
      WRITE(6,3825)XA4,YA4
 3825 FORMAT(1H ,'XA4 =',D22.14,' YA4 =',D22.14)
 3826 CONTINUE
      SUMXA4=SUMXA4+XA4
      SQXA4=SQXA4+(XA4**2)
      IWLSCN=IWLSCN+1
C
C******** DO WE HAVE TWENTY SCANS OF DATA **
C
      IF(IWLSCN .GE. 20)                                    GOTO 3830
      IF (IIN(3) .GE. 0)                                    GOTO 3827
      WRITE(6,111)IWLSCN
  111 FORMAT(1H ,'IWLSCN =',I5)
 3827 CONTINUE
      IF(NIX .LT. IHI)                                      GOTO 2130
      GOTO 3999
 3830 SQSD=((IWLSCN*SQXA4)-(SUMXA4**2))/((IWLSCN-1)*IWLSCN)
      SDWL=SQRT(SQSD)
      AVEWL=SUMXA4/IWLSCN
C
C******** STORE A4 PCM VAULE AVERAGE (AVEWL), STD. DEV. (SDWL) AND
C         CHANNEL IDENTIFIER (ICHAN). **
C
      CALL ZERO(NTABUF(1),50)
      NTABUF(1)=22
      INTNUM=AVEWL
      RFRAC=AVEWL-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(2)=INTNUM
      NTABUF(3)=IFRAC
      INTNUM=SDWL
      RFRAC=SDWL-INTNUM
      IFRAC=RFRAC*10000
      NTABUF(4)=INTNUM
      NTABUF(5)=IFRAC
      NTABUF(6)=JCHN(ICHAN)
      CALL NTRAN(12,1,50,NTABUF(1))
 3831 CALL NTRAN(12,15,LSTAT)
```

```
      IF(LSTAT+1)3832,3831,3833
 3832 LULU=12
      WRITE(6,4011)LULU,LSTAT
                                                        GOTO 4500
 3833 IWLHDR=0
      IWLFLG=1
      IF (IIN(3) .GE. 0)                                GOTO 1007
      WRITE(6,3835)SDWL,AVEWL
 3835 FORMAT(1H ,'SDWL =',F15.8,' AVEWL =',F15.8)
 1007 CONTINUE
      ICHAN=0
      IWLSCN=0
      SUMYA4=0.
      SQYA4=0.
      ITEST=0
      SNCHAN=0.
      SNA4=0.
      SUMXY=0.
      SUMX2Y=0.
      SUMX3=0
      SUMX4=0
      JCMPLT(2)=1
      NFSCN=0
      NX=0
      NSAMPL=0
      NIX=0
      IPSOK=0
      IWLHDR=0
      IWLFLG=1
      CALL ZERO(NCHAN(1),685)
                                                        GOTO 3999
 3836 CALL ZERO(NTABUF(1),50)
      NTABUF(1)=22
      CALL NTRAN(12,1,50,NTABUF(1))
 3837 CALL NTRAN(12,15,LSTAT)
      IF(LSTAT+1)3838,3837,1007
 3838 LULU=12
      WRITE(6,4011)LULU,LSTAT
                                                        GOTO 4500
 3850 CALL AMOV(MEAS(LOCA3),NCHAN(NSAMPL),NX)
      GOTO(2185,2450,2600)IBACK
 3900 CALL AMOV(MEAS(LOCA5),NCHAN(NSAMPL),NX)
      GOTO(2185,2450,2600)IBACK
 3950 CALL AMOV(MEAS(LOCA2),NCHAN(NSAMPL),NX)
      GOTO(2185,2450,2600)IBACK
 3999 IF (IIN(3) .GE. 0)                                GOTO 4100
      WRITE(6,4009)
 4100 CALL SEG('DK1:QAEXEC.GGG',0,0)
 4000 FORMAT(1H ,'LMTLO = ',I5)
 4001 FORMAT(1H ,'LMTHI = ',I5)
 4002 FORMAT(1H ,'IPK4 =',I5)
 4003 FORMAT(1H ,'IRAN =',I5)
 4004 FORMAT(1H ,'ITOP =',I5)
 4005 FORMAT(1H ,'JHI =',I5)
 4006 FORMAT(1H ,'KLO =',I5)
 4007 FORMAT(1H ,'INUM =',I5)
```

```
 4009 FORMAT(1H ,'LEAVING WVLPRO')
 4010 FORMAT(1H ,'A4 VALUE ',I4,'Y-AXIS:ACTUAL VALUE ',I4,
     *        'Y-AXIS:ESTIMATED VALUE ',D22.14)
 4011 FORMAT(1H0,'NTRAN WRITE ERROR ON UNIT',I4,' STATUS WORD',I4)
 4500 CALL SEGRET
      END

ROUTINES CALLED:
SEG    , AMOV   , ZERO   , STOHMS, PDUMP  , SQRT   , NTRAN
SEGRET

SWITCHES = /ON,/SU,/CO

BLOCK         LENGTH
MAIN.    4090    (017764)*
PRNDX    54      (000154)
ANDAT    91      (000266)
A4CHNL   685     (002532)
DCDATA   3993    (017462)
WVLDAT   4       (000010)
RESDAT   708     (002610)
HISDAT   8       (000020)
INPUT    158     (000474)
INTNDX   5       (000012)
SIXSV    1       (000002)
TIMES    522     (002024)

**COMPILER ----- CORE**
   PHASE        USED  FREE
DECLARATIVES 04765 12501
EXECUTABLES  05431 11835
ASSEMBLY     03502 16681
```



```
#
$EOD
$FI
TIME:-17:49:09
```

```
$JOB LINKER[300,006]
DATE:-27-JUL-76
TIME:-17:49:14
$RUN LINK
LINK V11A01
#DK1:DCMBUF.DA1,LP:,DK1:ST.STB<DK1:DCMBUF.OBJ[300,006]/B:140000/E
```

```
LOAD MAP  DCMBUF.DA1              17:49:24  27-JUL-76

TRANSFER ADDRESS: 000001
LOW LIMIT: 140000
HIGH LIMIT: 157462
**********
MODULE  DATA.
SECTION         ADDRESS SIZE
<. ABS.>        000000  000000
<      >        140000  000000
<DCDATA>        140000  017462
```

```
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:QA191H.LDA,LP:,DK1:ST.STB<DK1:     ST.STB[300,006]
#                                      DK1:QA191H.OBJ[300,006]
#                                      DK1:GLOBLS.OBJ[300,006]
#                                      DK0:  ABEC.OBJ[100,100]
#                                      DK0:   CVP.OBJ[100,100]
#                                      DK0: FIELD.OBJ[100,100]
#                                      DK0:   FLD.OBJ[100,100]
#                                      DK0:JDATE2.OBJ[100,100]
#                                      DK0:  MASK.OBJ[100,100]
#                                      DK0:   MSK.OBJ[100,100]
#                                      DK0:NTRAND.OBJ[100,100]
#                                      DK0:SEGOUT.OBJ[100,100]
#                                      DK0:SEGRES.OBJ[100,100]
#                                      DK0:SETEMT.OBJ[100,100]
#                                      DK0:STALPH.OBJ[100,100]
#                                      DK0:STOHMS.OBJ[100,100]
#                                      DK0:FTNLIB.OBJ[1,1]/L/T:57776/E
```

```
LOAD MAP  QA101H.LDA                 17:50:14  27-JUL-76

TRANSFER ADDRESS: 024372
LOW LIMIT: 024372
HIGH LIMIT: 057776
**********
MODULE  DATA.
SECTION              ADDRESS SIZE
<. ABS.>             000000  000000
<      >             140000  000000
<DCDATA>             140000  017462
**********
MODULE  MAIN.
SECTION              ADDRESS SIZE
<      >             024372  000550
         MAIN.       024372
<A4CHNL>             025142  002532
<ANDAT >             027674  000266
<CALRT >             030162  000006
<CMPLET>             030170  000004
<HVLDAT>             030174  000010
<LASFRM>             030204  000010
<HISDAT>             030214  000020
<BIADAT>             030234  000114
<DCARGN>             030350  000070
<ERROR >             030440  000050
<FLDAT >             030510  000010
<INPUT >             030520  000474
<INTNDX>             031214  000012
<PRNDX >             031226  000154
<QADAT >             031402  000012
<RECPTR>             031414  000002
<RESDAT>             031416  002620
<SAVE  >             034236  000102
<SIXSV >             034340  000002
<TIMES >             034342  002024
<TITLES>             036366  000100
**********
MODULE  GLOBLS
SECTION              ADDRESS SIZE
<      >             036466  000000
**********
MODULE  CNV          730405
SECTION              ADDRESS SIZE
<      >             036466  000706
         .ABEC       036466
**********
MODULE  CVP          721227
SECTION              ADDRESS SIZE
<      >             037374  000136
         CVP         037374              SRCVP    037374
**********
MODULE  FIELD        721227
SECTION              ADDRESS SIZE
<      >             037532  000124
         PACK        037532              UNPACK   037604
**********
MODULE  FLD          730405
SECTION              ADDRESS SIZE
<      >             037656  000564
         .FLD        037656
**********
MODULE  JDATE2
```



```
SECTION                ADDRESS  SIZE
<         >            040442   001456
          JDATE2       040442
**********
MODULE    MASK         730108
SECTION                ADDRESS  SIZE
<         >            042120   000030
          MASK         042120
**********
MODULE    MSK          730405
SECTION                ADDRESS  SIZE
<         >            042150   000140
          .MASK        042150
**********
MODULE    NTRAND       750106
SECTION                ADDRESS  SIZE
<         >            042310   001330
          BLKITS       043472            BLKMXS    043430              FLBLKS   043416
          LNKTBS       042310            LOGRCS    043534              LUSTAT   043576
          SFBLKS       043140            TRNTBS    042624
**********
MODULE    SEGOUT
SECTION                ADDRESS  SIZE
<         >            043640   000264
          REMAP        044062            SEGOUT    043640
**********
MODULE    SEGRES
SECTION                ADDRESS  SIZE
<         >            044124   000360
          NEWMMC       044246            SAVE1     044146              SAVE2    044150
          SEG          044152            SEGEND    044356              SEGRET   044370
**********
MODULE    SETEMT       730115
SECTION                ADDRESS  SIZE
<         >            044504   000162
          SETEMT       044504
**********
MODULE    STALPH       721209
SECTION                ADDRESS  SIZE
<         >            044666   000510
          STASC        045216            STBCD     045306              STEBC    045124
          ST.F10       045022            ST.G10    045056              ST.M14   045016
          ST.M15       045052            ST.ONE    045076              ST.P10   044676
          ST.TWO       045106            ST.32K    044666              ST.64K   045116
**********
MODULE    STOHMS
SECTION                ADDRESS  SIZE
<         >            045376   000132
          STOHMS       045376
**********
MODULE    SPRP01
SECTION                ADDRESS  SIZE
<         >            045530   000012
          SPOPP3       045530
**********
MODULE    SPOI02
SECTION                ADDRESS  SIZE
<         >            045542   000004
          SPOP1        045542            SPOP2     045542
**********
MODULE    SPOP03
SECTION                ADDRESS  SIZE
<         >            045546   000020
          SPOP3        045562            SPOP4     045546              SPOP4A   045554
          SPOP4B       045550            SPOP5     045546
**********
```

```
MODULE  $GET03
SECTION              ADDRESS SIZE
<       >            045566  000020
         $GET1       045602               $GET2    045602            $GET3    045576
         $GET4       045566               $GET5    045566
**********
MODULE  $PUT03
SECTION              ADDRESS SIZE
<       >            045606  000012
         $PUT1       045614               $PUT2    045614            $PUT3    045612
         $PUT4       045606               $PUT5    045606
**********
MODULE  $ASP04
SECTION              ADDRESS SIZE
<       >            045620  000022
         $ASP        045620               $POPP1   045624            $POPP2   045624
         $POPR1      045636               $POPR2   045636
**********
MODULE  $ADI03
SECTION              ADDRESS SIZE
<       >            045642  000014
         $ADI        045642
**********
MODULE  $CMI01
SECTION              ADDRESS SIZE
<       >            045656  000004
         $CMI        045656
**********
MODULE  $SBI01
SECTION              ADDRESS SIZE
<       >            045662  000004
         $SBI        045662
**********
MODULE  $TRX02
SECTION              ADDRESS SIZE
<       >            045666  000044
         $TRX        045666
**********
MODULE  $TR01
SECTION              ADDRESS SIZE
<       >            045732  000004
         $TR         045732
**********
MODULE  $ADJ03
SECTION              ADDRESS SIZE
<       >            045736  000040
         $ADJ        045736
**********
MODULE  $TRT02
SECTION              ADDRESS SIZE
<       >            045776  000014
         $TRTST      045776
**********
MODULE  $SVP01
SECTION              ADDRESS SIZE
<       >            046012  000010
         $SVP        046012
**********
MODULE  $SVE01
SECTION              ADDRESS SIZE
<       >            046022  000004
         $SVE        046022
**********
MODULE  $ED005
SECTION              ADDRESS SIZE
<       >            046026  000034
```

```
        $ENDDD  046026
**********
MODULE  $GLE01
SECTION           ADDRESS SIZE
<       >         046062  000032
        $EQ       046072            $GE     046104          $GT     046102
        $LE       046062            $LT     046064          $NE     046110
**********
MODULE  $LC002
SECTION           ADDRESS SIZE
<       >         046114  000076
        $LCO      046114
**********
MODULE  $IC002
SECTION           ADDRESS SIZE
<       >         046212  000272
        $ICO      046220            $OCO    046212
**********
MODULE  $DC004
SECTION           ADDRESS SIZE
<       >         046504  001614
        $DCO      046516            $ECO    046546          $FCO    046512
        $GCO      046504
**********
MODULE  $LCI01
SECTION           ADDRESS SIZE
<       >         050320  000076
        $LCI      050320
**********
MODULE  $ICI02
SECTION           ADDRESS SIZE
<       >         050416  000304
        $ICI      050424            $OCI    050416
**********
MODULE  $DCI01
SECTION           ADDRESS SIZE
<       >         050722  001400
        $DCI      050730            $RCI    050722
**********
MODULE  $IUD02
SECTION           ADDRESS SIZE
<       >         052322  000042
        $IOUD     052322
**********
MODULE  $IRD02
SECTION           ADDRESS SIZE
<       >         052364  000062
        $IORD     052364
**********
MODULE  $IOF06
SECTION           ADDRESS SIZE
<       >         052446  000216
        $IOERR    052564            $IOF    052446          $IOFX   052536
**********
MODULE  $ST003
SECTION           ADDRESS SIZE
<       >         052664  000024
        $STOP     052664
**********
MODULE  $NAM07
SECTION           ADDRESS SIZE
<       >         052710  000110
        $NAM      052710            $RET    052766
**********
MODULE  $MOD02
SECTION           ADDRESS SIZE
```

```
<         >          053020   000040
          MOD        053020
**********
MODULE  $INT02
SECTION              ADDRESS  SIZE
<         >          053060   000026
          IDINT      053060              INT       053060
**********
MODULE  $FLT02
SECTION              ADDRESS  SIZE
<         >          053106   000020
          FLOAT      053106
**********
MODULE  $AMD02
SECTION              ADDRESS  SIZE
<         >          053126   000056
          AMOD       053126
**********
MODULE  $TSI03
SECTION              ADDRESS  SIZE
<         >          053204   000042
          $TSB       053224              $TSD      053204              $TSI      053230
          $TSR       053210
**********
MODULE  $SBS08
SECTION              ADDRESS  SIZE
<         >          053246   000106
          $SBS1      053246              $SBS2     053254              $SBS3     053262
**********
MODULE  $MLI05
SECTION              ADDRESS  SIZE
<         >          053354   000024
          $MLI       053354
**********
MODULE  $MLR05
SECTION              ADDRESS  SIZE
<         >          053400   000012
          $MLR       053400
**********
MODULE  $IR04
SECTION              ADDRESS  SIZE
<         >          053412   000016
          $IC        053412              $ID       053412              $IR       053416
**********
MODULE  $DVR08
SECTION              ADDRESS  SIZE
<         >          053430   000014
          $DVR       053430
**********
MODULE  $DVI03
SECTION              ADDRESS  SIZE
<         >          053444   000026
          $DVI       053444
**********
MODULE  $ANT03
SECTION              ADDRESS  SIZE
<         >          053472   000042
          AINT       053472              $INTR     053514
**********
MODULE  $ADR04
SECTION              ADDRESS  SIZE
<         >          053534   000016
          $ADR       053540              $SBR      053534
**********
MODULE  $RI04
SECTION              ADDRESS  SIZE
```

```
<        >       053552  000016
         $DI     053552           $RI      053556
***********
MODULE   $FPR01
SECTION          ADDRESS SIZE
<        >       053570  000126
         $FPERR  053570
***********
MODULE   $POL07
SECTION          ADDRESS SIZE
<        >       053716  000004
         $FPU    053722           $POLSH   053716          $V20A    053716
***********
MODULE   $PPR04
SECTION          ADDRESS SIZE
<        >       053722  000020
         $POPR3  053734           $POPR4   053722          $POPR5   053722
***********
MODULE   $PHR07
SECTION          ADDRESS SIZE
<        >       053742  000012
         $PSHR1  053750           $PSHR2   053750          $PSHR3   053746
         $PSHR4  053742           $PSHR5   053742
***********
MODULE   $ERR15
SECTION          ADDRESS SIZE
<        >       053754  001040
         $ERR    053754           $ERRA    053764          $RTS     054154
***********
MODULE   $RDM03
SECTION          ADDRESS SIZE
<        >       055014  000140
         $RANDM  055014
***********
MODULE   $TRC06
SECTION          ADDRESS SIZE
<        >       055154  000276
         $TRCBK  055154
***********
MODULE   $OTV04
SECTION          ADDRESS SIZE
<        >       055452  000032
         $ERRWK  055466           $EXSW    055464          $NAMC    055460
         $OTSV   055452           $SEQC    055462
***********
MODULE   $ERC08
SECTION          ADDRESS SIZE
<        >       055504  000124
         $ERRC   055506           $ERRF    055616
***********
MODULE   $EXT05
SECTION          ADDRESS SIZE
<        >       055630  000034
         EXIT    055630           $EXIT    055630
***********
MODULE   $CLP01
SECTION          ADDRESS SIZE
<        >       055664  000072
         $CLSUP  055664
***********
MODULE   $FDV05
SECTION          ADDRESS SIZE
<        >       055756  000040
         $FNDEV  055756
***********
MODULE   $CLS04
```

```
SECTION            ADDRESS SIZE
<        >         056016  000104
        $CLOSE     056016              $CLSE     056052
**********
MODULE   $IST05
SECTION            ADDRESS SIZE
<        >         056122  000156
        $IOSET     056122
**********
MODULE   $DVB09    750106
SECTION            ADDRESS SIZE
<        >         056300  001230
        $DEVTB     056322
**********
MODULE   $IBF02
SECTION            ADDRESS SIZE
<        >         057530  000246
        $IOBUF     057532
**********
**********
UNDEFINED REFERENCES
.SHIFT
```

```
W322 000000
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:CONDRV,FFF<DK1:     ST,STB[300,006]
#                      DK1:CONDRV,OBJ[300,006]
#                      DK1:BLKDAT,OBJ[300,006]
#                      DK0:CONINP,OBJ[100,100]/CC
#                      DK0:HMSTOS,OBJ[100,100]
#                      DK0:FTNLIB,OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 116334
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:QAEXEC,GGG<DK1:     ST,STB[300,006]
#                          DK1:QAEXEC,OBJ[300,006]
#                          DK0:  AMOV,OBJ[100,100]
#                          DK1:DATDCM,OBJ[300,006]
#                          DK0: NTRAN,OBJ[100,100]
#                          DK0:  ZERO,OBJ[100,100]
#                          DK0:FTNLIB,OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 106450
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:ERRDRV,HHH<DK1:     ST,STB[300,006]
#                      DK1:ERRDRV,OBJ[300,006]
#                      DK0:ERRPRC,OBJ[100,100]
#                      DK0: NTRAN,OBJ[100,100]
#                      DK0:TERMIT,OBJ[100,100]
#                      DK0:FTNLIB,OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 113700
LINK V11A01
#
$EOD
$RUN LINK
.INK V11A01
#DK1:DCMDRV,III<DK1:     ST,STB[300,006]
#                      DK1:DCMDRV,OBJ[300,006]
#                      DK0:DCOM2N,OBJ[100,101]/CC
#                      DK0: NTRAN,OBJ[100,100]
#                      DK0:   TPI,OBJ[100,100]
#                      DK0:   DFD,OBJ[100,101]
#                      DK0:FTNLIB,OBJ[1,1]/L/B:60000/E

'RANSFER ADDRESS: 060000
.OW LIMIT: 060000
HIGH LIMIT: 133516
```

```
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:FLDPRO.JJJ<DK1:     ST.STB[300,006]
#                          DK1:FLDPRO.OBJ[300,006]
#                          DK0: NTRAN.OBJ[100,100]
#                          DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 070710
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:BIAPRO.KKK<DK1:     ST.STB[300,006]
#                          DK1:BIAPRO.OBJ[300,006]
#                          DK0:  ZERO.OBJ[100,100]
#                          DK0:  AMOV.OBJ[100,100]
#                          DK0: NTRAN.OBJ[100,100]
#                          DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 106360
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1: ANPRO.LLL<DK1:     ST.STB[300,006]
#                          DK1: ANPRO.OBJ[300,006]
#                          DK0:  AMOV.OBJ[100,100]
#                          DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 107476
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:WVLPRO.MMM<DK1:     ST.STB[300,006]
#                          DK1:WVLPRO.OBJ[300,006]
#                          DK0:  AMOV.OBJ[100,100]
#                          DK0:  ZERO.OBJ[100,100]
#                          DK0: NTRAN.OBJ[100,100]
#                          DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 112440
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:RAMPRO.NNN<DK1:     ST.STB[300,006]
#                          DK1:RAMPRO.OBJ[300,006]
#                          DK0:  ZERO.OBJ[100,100]
#                          DK0:  AMOV.OBJ[100,100]
#                          DK0: NTRAN.OBJ[100,100]
```

```
#                               DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 114634
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:RESPRO.OOO<DK1:     ST.STB[300,006]
#                           DK1:RESPRO.OBJ[300,006]
#                           DK0:  AMOV.OBJ[100,100]
#                           DK0:  ZERO.OBJ[100,100]
#                           DK0: NTRAN.OBJ[100,100]
#                           DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 125074
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:TMLOOP.PPP<DK1:     ST.STB[300,006]
#                           DK1:TMLOOP.OBJ[300,006]
#                           DK0:  AMOV.OBJ[100,100]
#                           DK0:  ZERO.OBJ[100,100]
#                           DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 070640
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1: QASUM.QQQ<DK1:     ST.STB[300,006]
#                           DK1: QASUM.OBJ[300,006]
#                           DK0:  AMOV.OBJ[100,100]
#                           DK0:  ZERO.OBJ[100,100]
#                           DK0: NTRAN.OBJ[100,100]
#                           DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 132122
LINK V11A01
#
$EOD
$RUN LINK
LINK V11A01
#DK1:DECRIP.RRR<DK1:     ST.STB[300,006]
#                           DK1:DECRIP.OBJ[300,006]
#                           DK1:DATDCM.OBJ[300,006]
#                           DK0: NTRAN.OBJ[100,100]
#                           DK0:DCRIPT.OBJ[100,101]/CC
#                           DK0:FTNLIB.OBJ[1,1]/L/B:60000/E

TRANSFER ADDRESS: 060000
LOW LIMIT: 060000
HIGH LIMIT: 122622
LINK V11A01
```

```
$JOB FLOADR[300,006]
DATE:-27-JUL-76
TIME:-18:01:15
$RUN FLOAD[100,100]
#/MMC:AAA
#/DATA
#/EOD
#DK1:CONDRV,FFF
#DK1:ERRDRV,HHH
#/END
#/MMC:BBB
#/DATA
#DK1:DCMBUF,DA1
#/EOD
#DK1:QAEXEC,GGG/NP:1/NB:1,DK1:DCMBUF,DA1
#DK1:ERRDRV,HHH
#DK1:DCMDRV,III
#DK1:FLDPRO,JJJ
#DK1:BIAPRO,KKK
#DK1: ANPRO,LLL
#DK1:WVLPRO,MMM
#DK1:RAMPRO,NNN
#DK1:RESPRO,OOO
#DK1:TMLOOP,PPP
#DK1: GASUM,QQQ
#DK1:DECRIP,RRR
#/END
#/EXIT

$EOD
$FI
TIME:-18:01:56
```

APPENDIX A

TAPE FORMATS

APPENDIX A
INTERMEDIATE NON-IMAGERY DATA TAPE FORMAT
FOR THE EARTH RESOURCES PRODUCTION PROCESSOR

I. INTRODUCTION

The purpose of this memorandum is to clarify and formalize a format for the intermediate non-imagery data tape for the Earth Resources Production Processor Project. This intermediate tape will be the data communication link between the preprocessor and the data processor. The format selected resulted from verbal agreements between Carl Lanham, Roy Jones, Pete Lucas and Paul Chen of the Preprocessing Group of PHO, and Charles Wilson and Theda Cook of the Data Processing Group of PHO. A continuous data format, which is self descriptive, compact, and general purpose was selected. This format will standardize the output of the preprocessor and the input to the scientific computational programs.

II. GUIDELINES FOR TAPE FORMAT

The following guidelines were established for the intermediate tape format:

a. The tape will be a 9-track tape with a packing density of 800 BPI in odd parity or a high density data tape.

b. Data requiring alphanumeric character representation will use ASCII representation left justified in the field with 8-bits allowed for each character. Character strings will appear on the tape in the order of natural occurrence. For example, DESCRIPT appears with D first followed by E, S, C, R, I, P, T, in that order.

c. Integer data will be right justified in specified fields with the least significant bit on the right of the field. All data is written in byte pairs with the right most byte of a pair written first.

d. The intermediate tape will contain two types of information files:

   1) Descriptor file(s)
   2) Data files

e. The tape will be written in variable length physical records with fixed length physical records of 3000 8-bit bytes in the descriptor file and physical records not to exceed 3000 8-bit bytes in the data files.

f. With the three exceptions addressing units in the data file will be 12-bit increments beginning with address 0 for the first 12 bits of information. BADD and RCDLGH, defined in Section III, will be in 8-bit increments and LGPF, also defined in Section III, will be in 16 bit increments.

g. All descriptor file(s) will precede the data file on the tape.

h. Each field in the descriptor file will be in multiples of 8 bits with a 64 bit maximum.

i. There will be one descriptor file for each data source on the tape; a source corresponds to one track of a 14-track EOAP, EREP or ERTS data tape.

j. Each descriptor file will have a variable number of physical records depending on the number of data parameters in the data records it describes.

k. A descriptor file will contain at least one identifier block for each parameter in the data records it describes. An identifier block names and locates a parameter within data records.

l. The end of a descriptor file will be indicated by the second 64-bit field of binary zeros following the last identifier block. There will be no end of file after the descriptor file.

m. There will be one data record per physical record. Each data record will be divided into a header block and a data block. The header block will consist of date, time, synchronization, mode, and indexing information to be used for asynchronous data retrieval. The data block will contain the data measurements. The maximum length for a data record will be 3000 8-bit fields.

n. The identifier blocks in the first physical record of the descriptor file will describe the parameters in the header blocks of the associated data files. These identifier blocks will be referred to as header identifier blocks. The last header identifier block will be followed by 64 binary zeros.

o. The remainder of the first physical record, not required by header identifier blocks, and subsequent physical records in the descriptor file will contain identifier blocks to be referred to as measurement identifier blocks, which will describe parameters in the data blocks of the associated data files. If a complete identifier block cannot be contained at the end of a record, the end of that record will be filled with binary zeroes. An identifier block will not be split between two physical records.

p. Each field in the data file will be aligned on 12-bit boundaries. A data file will contain data from only one source, a source being one track of EOAP, EREP, or ERTS data tape. (Theoretically there could be data from up to four data sources in the same data file. In this case there would be up to four header blocks and four associated data blocks in the file. This capability is useful in the merging of data streams from up to four sources, but it is not applicable to the intermediate data tapes for this project.)

q. Since this format is modeled after the Serial Data Tape format in the Earth Resources Data Format Control Book, it contains unused fields. These fields will be filled with binary zeros.

r. The last data file on the tape will be followed by an end-of-file mark.

III. DESCRIPTOR FILE FORMAT

The descriptor file(s) will identify the type(s) and format(s) of the data records on the tape. The following information will be contained in each descriptor record which will begin with the ASCII representation of "DESCRIPT" in the first 64 bits of the file:

a. Date of data recording.

b. Format number as defined in the Appendix to correlate data records to the proper descriptor file for formatting.

c. Identification number.

d. Data structure information as received from the 14-track tape.

e. An identifier block for each parameter in the data records for the data source.

An identifier block will define the ASCII name of a data parameter, the field position and length, associated sync words, and subcommutative and supercommutative indexing information for the parameter. There will be two types of identifier blocks: (1) header identifier blocks which define parameters in the header blocks of data records and (2) measurement identifier blocks which define parameters (or measurements) in the data blocks of data records.

The general format of a core buffer containing a descriptor file is outlined in Figure 1. The symbols used in Figure 1 are defined below. The definitions using byte addresses refer to addresses in a core buffer with the byte pairs in reverse order from sequential byte positions on the tape.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
DESCRIPT	ASCII(A)	0-7	ASCII representation of "DESCRIPT"
MONTH	A	8-9	Month of data recording, 01-12, in ASCII
YEAR	A	10-11	The last two digits of the year the data was recorded
MISSION	A	12-15	Mission number
FORMAT	Binary Integer(BI)	16-17	Format number used to correlate the descriptor file with the proper data files and to indicate the recording format of the data source. Values of FORMAT are defined in the Appendix.

FIGURE 1

BYTE NUMBER		15 — 8	7 — 0
0	1	E	D
2	3	C	S
4	5	I	R
6	7	T	P
8	9	MONTH	
10	11	YEAR	
12	13	MISSION NUMBER	
14	15	MISSION NUMBER (CONTINUE)	
16	17	FORMAT NUMBER	
18	19	ID NUMBER	
20	21	ID NUMBER (CONTINUE)	
22	23	ID NUMBER (CONTINUE)	
24	25	NPFR	
32	33	NP	
40	41	LGPF	
48	49	BADD	
56	57	NBW	
64	65	BPS	
66	67	BPS (CONTINUE)	

FIGURE 1 (CONTINUED)

BYTE NUMBER		15 … 8	7 … 0
80	81	RCDLGH	
82	83	RCDLGH (CONTINUE)	
84	85	NHE	
86	87	NHE (CONTINUE)	
88	89	RPS INT	
90	91	RPS INT (CONTINUE)	
92	93	RPS FRAC	
94	95	RPS FRAC (CONTINUE)	
118	119		
120	121	HEADER IDENTIFIER	
122	123	HEADER IDENTIFIER (CONTINUE)	
124	125	HEADER IDENTIFIER (CONTINUE)	
126	127	HEADER IDENTIFIER (CONTINUE)	
128	129		
130	131	POS	
132	133	MAX1	
134	135	INCR	
136	137	NB	LSB
138	139	NF	
140	141	FW	DR
142	143		SYNC
144	145	HEADER IDENTIFIER	
146	147	HEADER IDENTIFIER (CONTINUE)	

FIGURE 1 (CONTINUED)

15 8 7 0

HEADER IDENTIFIER (CONTINUE)	
HEADER IDENTIFIER (CONTINUE)	
POS	
MAX1	
INCR	
NB	LSB
NF	
FW	DR
	SYNC
HEADER IDENTIFIER	
HEADER IDENTIFIER (CONTINUE)	
HEADER IDENTIFIER (CONTINUE)	
HEADER IDENTIFIER (CONTINUE)	
POS	
MAX1	
INCR	
NB	LSB
NF	
FW	DR
	SYNC
0—0	
0—0	
0—0	
0—0	
MEASUREMENT IDENTIFIER	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	

INDICATES END OF HEADER IDENTIFIER BLOCKS

FIGURE 1 (CONTINUED

<table>
<tr><th>15 8</th><th>7 0</th></tr>
<tr><td colspan="2">PINO</td></tr>
<tr><td colspan="2">POS</td></tr>
<tr><td colspan="2">FSTIND</td></tr>
<tr><td colspan="2">INCR</td></tr>
<tr><td>NB</td><td>LSB</td></tr>
<tr><td colspan="2">NF</td></tr>
<tr><td>FW</td><td>DR</td></tr>
<tr><td>IR</td><td>SR</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">PINO</td></tr>
<tr><td colspan="2">POS</td></tr>
<tr><td colspan="2">FSTIND</td></tr>
<tr><td colspan="2">INCR</td></tr>
<tr><td>NB</td><td>LSB</td></tr>
<tr><td colspan="2">NF</td></tr>
<tr><td>FW</td><td>DR</td></tr>
<tr><td>IR</td><td>SR</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">MEASUREMENT IDENTIFIER (CONTINUE)</td></tr>
<tr><td colspan="2">PINO</td></tr>
<tr><td colspan="2">POS</td></tr>
<tr><td colspan="2">FSTIND</td></tr>
<tr><td colspan="2">INCR</td></tr>
<tr><td>NB</td><td>LSB</td></tr>
<tr><td colspan="2">NF</td></tr>
<tr><td>FW</td><td>DR</td></tr>
<tr><td>IR</td><td>SR</td></tr>
<tr><td colspan="2"></td></tr>
</table>

BYTE 3000

NOTE: If a complete identifier block cannot be contained at the end of a record, the end of that record will be filled with zeros and the identifier block in question will begin at the beginning of the next record.

FIGURE 1 (CONTINUED)

15	7 0
MEASUREMENT IDENTIFIER	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	
PINO	
POS	
FSTIND	
INCR	
NB	LSB
NF	
FW	DR
IR	SR
.	
.	
.	
.	
.	
MEASUREMENT IDENTIFIER	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	
MEASUREMENT IDENTIFIER (CONTINUE)	
PINO	
POS	
FSTIND	
INCR	
NB	LSB
NF	
FW	DR

3000 BYTES PER RECORD
1 RECORD CONTAINS SPACE FOR 125 ENTRIES IF ALL ARE NOT FILLED THEN 64 BINARY ZEROS SIGNIFY END OF RECORD

FIGURE 1 (CONTINUED)

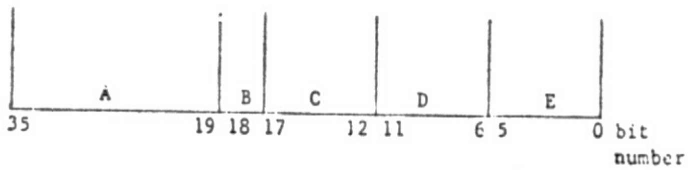
15 8	7 0	
IR	SR	
0	0	END OF MEASUREMENT IDENTIFIER BLOCK
0	0	
0	0	
0	0	

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
ID	A	18-23	ID number
NPFR	B.I.	24-25	Number of prime frames (as defined for the intermediate tape format) per data record.
NP	B.I.	32-33	Number of pins or data measurements per prime recording frame in original recording of data.
LGPF	B.I.	40-41	Length of storage required per prime frame in 16-bit fields.
BADD	B.I.	48-49	8-bit byte address of the beginning of the first prime frame of data in the record, e.g., the length of the header block of the data record (in 8-bit increments). BADD must be an even number. Zero fill will be added if necessary.
NBW	B.I.	56-57	Number of bits per word in original recording of data.
BPS	B.I.	64-67	Bit rate, in bits per second, of original recording of data.
RCDLGH	B.I.	80-83	Data record length in 8-bit increments.
NHE	B.I.	84-87	Number of header identifier blocks.
RPS	B.I.	88-95	Prime frame rate, in revolutions per second, of original recording of data, the first 32 bits are the integer portion and the second 32 bits are the fractional portion.
		96-119	Zero fill.
IDENT	A	0-7 (Relative to identifier block)	Header identifier block: The ASCII name of the header parameter, with trailing blanks to fill out the field. (Some header parameters will be assigned standard ASCII names in all descriptor files.) The maximum number of header identifier blocks is 120.
			End of the header identifier blocks: 64 zero bits.
	A	0-7	Measurement identifier block: The ASCII name of the measurement entry. (This will be the 8 character measurement number assigned to each measurement in the Earth Resources Data Format Control Book.)
			End of descriptor file: 64 zero bits
PINO	-	8-9	Header identifier block: Not applicable
PINO	B.I.	8-9	Measurement identifier block: The Pulse Code Modulated (PCM) word or pin number from which the measurement came, starting with pin number 0.
POS	B.I.	10-11	Header identifier block: The 12-bit byte address of the first byte of the data field containing the header parameter, relative to the first of the data record.
	B.I.	10-11	Measurement identifier block - subcommutated and main frame data: The 12-bit byte address of the first byte of the data field containing the measurement relative to the beginning of a prime frame.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
			Measurement identifier block - supercommutated data: The 12-bit byte address of the <u>first</u> entry for the measurement in a prime frame relative to the beginning of the prime frame.
MAXI	B.I.	12-13	Header identifier block: If IDENT is one of the INDEX fields, then this is the maximum value that the INDEX may have. (INDEX is a standard parameter in the header block of a data file for indexing subcommutated data. MAXI is the number of prime frames per cycle for the subcommutating rate associated with the INDEX.) Not applicable for any other header parameters.
	-	-	Measurement identifier block: Not applicable.
FSTIND	-	-	Header identifier block: Not applicable.
	B.I.	12-13	Measurement identifier block - subcommutated data: A number that associates a prime frame with a subcommutated measurement. If FSTIND, for a measurement, minus 1 equal the value in the INDEX table entry for a prime frame, the measurement occurs in the prime frame. (See definition of INDEX1, INDEX2, and INDEX4 in Section IV.) When the physical record contains an exact multiple of MAXI frames (MAXI is the number of prime frames in the cycle for the INDEX table), FSTIND associates a measurement with only the first prime frame in the the record, that has an INDEX entry equal to FSTIND minus 1, is the frame containing the first value of the measurement. For subsequent frames in the record, containing the measurement, FSTIND must be ignored and the measurements are found using INCR (defined below).
			Measurement identifier block - supercommutated and main frame data: Always 1.
INCR	B.I.	14-15	Header identifier block: The 12-bit byte address increment between successive fields containing the header parameter.
	B.I.	14-15	Measurement identifier block - subcommutated data: The prime frame increment to the next prime frame in which the measurement occurs.
			Measurement identifier block - supercommutated data: The 12-bit byte address increment to the next data field in the prime frame containing the measurement. The data field is defined by FW below. (The entries must be evenly spaced within each prime frame.)
NB	B.I.	16	Header identifier block: The number of bits per parameter value.
	B.I.	16	Measurement identifier block: The number of bits per measurement value.
LSB	B.I.	17	Header identifier block: The bit number of the least significant bit of the header parameter in the data field, starting with the right most bit in the data field as bit zero. (The data field is specified by FW which is defined below.)
	B.I.	17	Measurement identifier block: The bit number of the least significant bit of the measurement in the data field, starting with the right most bit in the data field as bit number zero.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
NF	B.I.	18-19	Header identifier block: The number of sequential fields of length defined by FW below. NF is used to indicate the number of entries in a table in the header block and it is usually equal to the number of prime frames per data record. (See the definition of INDEX1 and TIME1 in Section IV for an example of a table.)
	B.I.	18-19	Measurement identifier block - subcommutated and main frame data: Always 1.
			Measurement identifier block supercommutated data: The number of entries of the measurement per prime frame.
FW	B.I. B.I.	20 20	Header Identifier Block and Measurement Identifier Block:

A code to indicate the number of bits required to define a data field, which is always a multiple of 12 and always less than or equal to 48. FW=0 indicates a 12-bit field, FW=1 indicates a 24-bit field, FW=2 indicates a 36-bit field, and FW=3 indicates a 48 bit field. Using NB, LSB, and FW, a data field may be defined in multiples of 12 bits, but the subfields within the data fields may be defined ignoring the 12-bit boundaries. For example, consider the following data field.



Then NB, LSB, and FW are given for the parameters in the data field as follows:

Parameter IDENT	NB	LSB	FW
A	17	19	2
B	1	18	2
C	6	12	2
D	6	6	2
E	6	0	2

A, B, C, D, and E may be header parameters or measurements.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
DR	B.I.	21	Always equal to 1.
IR	-	-	Header identifier block: Not applicable.
	B.I.	22	Measurement identifier block - subcommutated data: Indicator of the index table associated with the measurement; 1, 2, 3, or 4 indicating INDEX1, INDEX2, INDEX3, or INDEX4, respectively. Measurement identifier block - supercommutated and main frame data: Always 0.
SYNC	B.I.	23	Header identifier block: A number, N, such that $0<N\le 15$ associating the proper SYNC table to the parameter. The SYNC table is a standard table in the header block of a data record. If the parameter being defined is a SYNC table, then N is the number of the SYNC table (1-15). For other parameters, then the relevant SYNC table is that one whose SYNC entry is equal to N. Measurement identifier block: Not applicable.
SR	-	-	Header identifier block: Not applicable.
	B.I.	23	Measurement identifier block: A number, N, associating the proper SYNC table to the measurement; the relevant SYNC table is that one whose SYNC entry is equal to N.

IV. DATA RECORD FORMAT

A data record will contain two major blocks: a header block, and a data block. There will be one data record in each physical record.

Header Block

The header block of a data record will consist of time, date, synchronization parameters, mode, and indexing parameters for subcommutated data. Some of the parameters in the header block will have standard ASCII names and descriptions: DAYS, YEAR, FMTNO, TIME1, SYNC1, SYNC2, ...SYNC15, INDEX1, INDEX2, INDEX3, and INDEX4. These parameters are defined as follows:

DAYS	B.I.	-	The day of the year the intermediate tape was generated.
YEAR	B.I.	-	The year the intermediate tape was generated.
FMTNO	B.I.	-	The format number of the descriptor file to be associated with the data record (see definition of FORMAT in Section III).
TIME1	B.I.	-	A time table containing a time in tenths of milliseconds associated with each prime frame in the data rec Days, hours, minutes, and seconds (in tenths of milliseconds) incremented from the zero hour of December 31 of the year before the year indicated in the descriptor file.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
INDEX1, INDEX2, INDEX3, and INDEX4	B.I.	-	Tables to be used for referencing subcommutated data in the data record. One INDEX table will be required for each asynchronous-subcommutating rate in the record. There is an entry in each INDEX table for each frame. The value in the INDEX table for a particular frame identifies the subcommutated measurement parameter in the frame; e.g., if in the measurement identifier block for a parameter IR = 1 (specifies INDEX1) and FSTIND = 4 (specifies a value of 3 in INDEX tabl then the measurement corresponding to the measurement identifier block will be found in those frames of the record that have a 3 in INDEX1. If the value in the INDEX table plus 1 is greater than MAXI in the header identifier block for the INDEX table, then the data synchronization pattern has not been recognized by the preprocessor and recording will begin in the INDEX table as soon as synchronization has been established. The length of the entries in the INDEX tables will be defined by NB in the respective header identifier blocks.
;1, SYNC2, . . . SYNC15	B.I.	-	Tables of status indicators for the data record. There will be one status indicator in each table for each prime frame. SYNC1 will correspond to TIME1, and it will have the following values indicating status from the timing subsystem: 7 - Timing subsystem indicated time code translated with no problems. 4 - Translator time discontinuity status line on 3 - Invalid time status line on, or if data from digital tape - parity error on read 1 - Both discontinuity and invalid status lines on 0 - No time read for this entry SYNC2, SYNC3, SYNC4 will be reserved; they will have nc application on the intermediate tape for this project. SYNC5 will be the prime frame synchronization indicator corresponding to TIME1. SYNC6, SYNC7, SYNC8 will be reserved; they will have no application on the intermediate tape for this project.

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
			SYNC9, SYNC10, SYNC11, and SYNC12 will be reserved to correspond to subcommutated data cycles indicated in INDEX1, INDEX2, INDEX3, and INDEX4, respectively. SYNC13, SYNC14, and SYNC15 will be used as required. The length of the individual entries in the SYNC tables will be defined by NB in the respective header identifier blocks. SYNC5 through SYNC15 will take on the following values or additional values as necessary when referencing data: 7 - The prime frame synchronization pattern was recognized with preprogrammed errors or confidence built up in the INDEX value. 5 - A one-bit slip was recognized in the prime frame synchronization pattern or medium confidence in the INDEX value. 3 - Low confidence in the INDEX value or if digital tape input-parity error on read 0 - Data bad or no data stored for this entry

Other header parameters may be defined as required.

Data Block

The data block of a data record will be composed of data frames. The data frames will be stored in time sequence, the number of time sequences per record being equal to the NPFR field in the descriptor file. Each time sequence, or consecutive frame, will correspond to consecutive entries in the TIME1, INDEX, and SYNC tables. The data in each data frame will be stored consecutively by PCM word number beginning with word 0 for the PCM word containing the sync indicator. The measurements will be packed in multiples of 12-bit fields within each prime frame. The packing format will be specified in the measurement identifier blocks. Refer to NB, SB, and FW in the measurement identifier blocks of the descriptor file as defined in Section III. Binary zeros will be used in a data field to force alignment on 12, 24, 36, or 48-bit boundaries. Alignment will be forced at the end of each prime frame

NON-IMAGERY DATA UNIVERSAL FORMAT

DATA RECORD (Raw Data)

Byte No.	Content	Description	
1-4	.	Frame Time for Block # 1	
5-6	.	A001-RRO	1st sample this block
.	.	.	
67-68	.	A001-RRO	32nd sample this block
69-70	.	A002-RRO	1st sample this block
.	.		
131-132	.	A002-RRO	32nd sample this block
133-134	.	A003-RRO	1st sample this block
	.		
195-196	.	A003-RRO	32nd sample this block
197-198	.	A004-RRO	1st sample this block
	.		
259-260	.	A004-RRO	32nd sample this block
261-262	.	A005-RRO	1st sample this block
	.		
323-324	.	A005-RRO	32nd sample this block
325-326	.	A006-RRO	1st sample this block
	.		
387-388	.	A006-RRO	32nd sample this block
389-390	.	A007-RRO	
391-392	.	A008-RRO	
393-394	.	A009-RRO	
395-396	.	A010-RRO	
397-398	.	A011-RRO	
399-400	.	A013-RRO	
401-402	.	A014-RRO	
403-404	.	A015-RRO	
405-406	.	D005-RRO	
407-408	.	D006-RRO	
409-410	.	A101-RRO	
411-412	.	A101-RRO	
413-414	.	A106-RRO	

SYMBOL	DATA TYPE	8-BIT BYTE LOCATION	DEFINITION
			SYNC9, SYNC10, SYNC11, and SYNC12 will be reserved to correspond to subcommutated data cycles indicated in INDEX1, INDEX2, INDEX3, and INDEX4, respectively.
			SYNC13, SYNC14 and SYNC15 will be used as required.
			The length of the individual entries in the SYNC tables will be defined by NB in the respective header identifier blocks.
			SYNC5 through SYNC15 will take on the following values or additional values as necessary when referencing data:
			7 - The prime frame synchronization pattern was recognized with preprogrammed errors or confidence built up in the INDEX value.
			5 - A one-bit slip was recognized in the prime frame synchronization pattern or medium confidence in the INDEX value.
			3 - Low confidence in the INDEX value or if digital tape input-parity error on read
			0 - Data bad or no data stored for this entry

Other header parameters may be defined as required.

Data Block

The data block of a data record will be composed of data frames. The data frames will be stored in time sequence, the number of time sequences per record being equal to the NPFR field in the descriptor file. Each time sequence, or consecutive frame, will correspond to consecutive entries in the TIME1, INDEX, and SYNC tables. The data in each data frame will be stored consecutively by PCM word number beginning with word 0 for the PCM word containing the sync indicator. The measurements will be packed in multiples of 12-bit fields within each prime frame. The packing format will be specified in the measurement identifier blocks. Refer to MB, LSB, and FW in the measurement identifier blocks of the descriptor file as defined in Section III. Binary zeros will be used in a data field to force alignment on 12, 24, 36, or 48-bit boundaries. Alignment will be forced at the end of each prime frame

NON-IMAGERY DATA UNIVERSAL FORMAT

DATA RECORD (Raw Data)

Byte No.	Content	Description	
1-4	.	Frame Time for Block # 1	
5-6	.	A001-RRO	1st sample this block
.	.	.	
67-68	.	A001-RRO	32nd sample this block
69-70	.	A002-RRO	1st sample this block
.	.		
131-132	.	A002-RRO	32nd sample this block
133-134	.	A003-RRO	1st sample this block
	.		
195-196	.	A003-RRO	32nd sample this block
197-198	.	A004-RRO	1st sample this block
	.		
259-260	.	A004-RRO	32nd sample this block
261-262	.	A005-RRO	1st sample this block
	.		
323-324	.	A005-RRO	32nd sample this block
325-326	.	A006-RRO	1st sample this block
	.		
387-388	.	A006-RRO	32nd sample this block
389-390	.	A007-RRO	
391-392	.	A008-RRO	
393-394	.	A009-RRO	
395-396	.	A010-RRO	
397-398	.	A011-RRO	
399-400	.	A013-RRO	
401-402	.	A014-RRO	
403-404	.	A015-RRO	
405-406	.	D005-RRO	
407-408	.	D006-RRO	
409-410	.	A101-RRO	
411-412	.	A101-RRO	
413-414	.	A106-RRO	

Byte No.	Content	Description
415-416	.	D007-RRO
417-418	.	A016-RRO
419-420	.	A017-RRO
421-422	.	A017-RRO
423-424	.	A018-RRO
425-426	.	A019-RRO
427-428	.	A020-RRO
429-430	.	A020-RRO
431-432	.	A023-RRO
433-434		A102-RRO
435-436		A103-RRO Last Sample Block #1
2615-2616		A103-RRO .Last Sample Block #6
2617-2880	0-0	Zero Fill

C-7

APPENDIX B

TABULATION FORMATS

LIST OF INPUT CARDS

CARD NUMBER	CARD COLUMNS 12345678901234567890123456789012345678901234567890123456789012345678901234567890							
1								
2	15	27	49	1	2	3	4	
3	S191H QUALITY TEST STEP NO. 3 ABCDEFGHIJKLMNOPQURSTUVWXYZ							
4	76	05	06	1	0	0	0	0
5	20	5	5	65				
6	14	7	26.		14	7	35.	
7	14	11	4.		14	11	30.	
8	14	12	10.		14	16	20.	
9	4	84.234			5	64.234		
10	5	95.234			6	75.234		
11	6	106.234			7	86.234		
12	7	117.234			8	97.234		
13	8	128.234			9	108.234		
14	9	139.234			10	119.234		
15	10	410.23			11	1210.23		
16	11	1511.23			12	1311.23		
17	12	1612.23			13	1412.23		
18	13	1612.23			13	1712.23		
19	13	1713.23			13	1714.23		
20	13	1715.23			13	1716.23		
21	13	1717.23			13	1718.23		
22	13	1719.23			13	1720.23		
23	13	1721.23			13	1722.23		
24	13	1723.23			13	1724.23		
25	13	1725.23			13	1726.23		
26	14	12	18.		14	16	20.	
27	2	25	2.222		2	40	2.222	
28	3	25	3.333		3	40	3.333	
29	4	25	4.444		4	40	4.444	
30	5	25	5.555		5	40	5.555	
31	14	11	5.		14	11	30.	
32	7	25	2.222		7	40	2.222	
33	8	25	3.333		8	40	3.333	
34	9	25	4.444		9	40	4.444	
35	10	25	5.555		10	40	5.555	
36	14	7	26.		14	7	35.	
37	2	25	10.		2	25	13.	
38	3	25	10.		3	25	13.	
39	4	25	10.		4	25	13.	
40	5	25	10.		5	25	13.	
41	14	14	0.		14	15	0.	
42	2	25	2.222		2	35	0.	
43	3	25	3.333		3	35	0.	
44	4	25	4.444		4	35	0.	
45	5	25	5.555		5	35	0.	
46	14	15	30.		14	16	30.	
47	2	35	0.		2	38	0.	
48	3	35	0.		3	38	0.	
49	4	35	0.		4	38	0.	
50	5	35	0.		5	38	0.	
51	14	12	15.		14	13	0.	
52	2	38	0.		2	40	2.222	
53	3	38	0.		3	40	3.333	
54	4	38	0.		4	40	4.444	
55	5	38	0.		5	40	5.555	
56	14	14	18.		14	14	17.	
57	2	25	2.222		2	35	0.	
58	3	25	3.333		3	35	0.	
59	4	25	4.444		4	35	0.	
60	5	25	5.555		5	35	0.	
	14	19	37.		14	18		

61											
62	2	35		0.	2	38		0.			
63	3	35		0.	3	38		0.			
64	4	35		0.	4	38		0.			
65	5	35		0.	5	38		0.			
66	14	11		4.	14	11		30.			
67	7	25		2,222	7	40		2,222			
68	8	25		3.333	8	40		3,333			
69	9	25		4,444	9	40		4,444			
70	10	25		5.555	10	40		5,555			
71	5	5	1020	956	610	685					
72	38	39	40	41	42	43	44				
73	2	3	4	5	6	7	8				
74	9	10	11	12	13	14	15				
75	3	4	5	6	7	8	9				
76	37	38	39	40	41	42	43				
77		1,33		0,00872		0,024		-478,110	385,124	0,0	0,0
78		.65		.006		0,059		324,985	362,562	341,144	0,0
79		.8		0,00925		0,030		521,188	361,328	-27,3428	0,0
80		12,3		1.0		0,202		-545,409	60.7470	0.0	0,0
81	600	650									
82	2	0	498	494	600	0	550	0			
83	500	0	700	0	1023	0	550	0			
84	900	0	650	200	185	165	535	520			
85	2	0									
86	915	865	675	625	75	25	325	275			
	925	875	400	350	640	520					

LIST OF STORED INPUT PARAMETERS

PROJECT 5191 QUALITY TEST STEP NO. 3 ABCDEFGHIJKLMNOPQURSTUVWXYZ

BASE JULIAN DATE 2442904.5

TIME INTERVALS (SECONDS)

NUMBER	START TIME	STOP TIME	NUMBER	START TIME	STOP TIME
1	5.0846000000000000D 04	5,0855000000000000D 04	2	5,1064000000000000D 04	5,1090000000000000D 04
3	5.1130000000000000D 04	5.1380000000000000D 04	4	1,488423400020599D 04	1,836423400020599D 04
5	1.854[illegible]23400020599D 04	2,202523400020599D 04	6	2,220623400020599D 04	2,568623400020599D 04
7	2,586723400020599D 04	2,934723400020599D 04	8	2,952823400020599D 04	3,300823400020599D 04
9	3.318923400020599D 04	3,666923400020599D 04	10	3,846023000000417D 04	4,033022999954224D 04
11	4.051122999954224D 04	4.399122999954224D 04	12	4,417222999954224D 04	4,765222999954224D 04
13	4.777222999954224D 04	4.783222999954224D 04	14	4,783322999954224D 04	4,783422999954224D 04
15	4.783522999954224D 04	4.783622999954224D 04	16	4,783722999954224D 04	4,783822999954224D 04
17	4.783922999954224D 04	4.784022999954224D 04	18	4,784122999954224D 04	4,784222999954224D 04
19	4.784322999954224D 04	4.784422999954224D 04	20	4,784522999954224D 04	4,784622999954224D 04
21	5.1130000000000000D 04	5.1380000000000000D 04	22	8,7022219998836520D 03	9,6022219998836520D 03
23	1.2303332999944690D 04	1.3203332999944690D 04	24	1,5904443999767300D 04	1,6804443999767300D 04
25	1.9505554999828340D 04	2.0405554999828340D 04	26	5.1065000000000000D 04	5,1090000000000000D 04
27	2.6702221999883650D 04	2.7602221999883650D 04	28	3,0303332999944690D 04	3,1203332999944690D 04
29	3.3904443999767300D 04	3.4304443999767300D 04	30	3.7505554999828340D 04	3,8405554999828340D 04
31	5.0846000000000000D 04	5.0855000000000000D 04	32	8.7100000000000000D 03	8,7130000000000000D 03
33	1.2310000000000000D 04	1.2313000000000000D 04	34	1,5910000000000000D 04	1,5913000000000000D 04
35	1.9510000000000000D 04	1.9513000000000000D 04	36	5,1240000000000000D 04	5,1300000000000000D 04
37	8.7022219998836520D 03	9.3000000000000000D 03	38	1,2303332999944690D 04	1,2900000000000000D 04
39	1.5904443999767300D 04	1.6500000000000000D 04	40	1,9505554999828340D 04	2,0100000000000000D 04
41	5.1330000000000000D 04	5,1390000000000000D 04	42	9,3000000000000000D 03	9,4800000000000000D 03
43	1.2900000000000000D 04	1.3080000000000000D 04	44	1,6500000000000000D 04	1,6680000000000000D 04
45	2.0100000000000000D 04	2.0280000000000000D 04	46	5,1135000000000000D 04	5,1180000000000000D 04
47	9,4800000000000000D 03	9,6022219998836520D 03	48	1,3080000000000000D 04	1,3203332999944690D 04
49	1.6680000000000000D 04	1.6[illegible]4439997673[illegible]D 04	50	2,0280000000000000D 04	2,0405554999828340D 04
51	5.1252000000000000D 04	5.1257000000000000D 04	52	8,7022219998836520D 03	9,3000000000000000D 03
53	1.2303332999944690D 04	1.2900000000000000D 04	54	1,5904443999767300D 04	1,6500000000000000D 04
55	1.9505554999828340D 04	2.0100000000000000D 04	56	5,1337000000000000D 04	5,1342000000000000D 04
57	9.[illegible]000000000000000D 03	9.4800000000000000D 03	58	1,2900000000000000D 04	1,3080000000000000D 04

59	1.650000000000000D 04	1.668000000000000D 04	60	2.010000000000000D 04	2.028000000000000D 04
61	5.106400000000000D 04	5.109000000000000D 04	62	2.673222199988365D 04	2.760222199988365D 04
63	3.030333299994469D 04	3.120333299994469D 04	64	3.390444399976730D 04	3.480444399976730D 04
65	3.750555499982834D 04	3.840555499982834D 04			

INDEX	PARAMETER	VALUE	INDEX	PARAMETER	VALUE	INDEX	PARAMETER	VALUE	INDEX	PARAMETER	VALUE
1	SENSOR	15	2	REC. FORM.	27	3	MISSION	99	4	FLIGHT NO.	1
5	SITE NO.	2	6	LINE NO.	3	7	RUN NO.	4			
8	YEAR	76	9	MONTH	5	10	DAY	6	11	TAB OPTION	1
12	FL DEL OPT	0	13	HRS. DEL	0	14	MINS. DEL	0	15	SECS. DEL	0
16	OVRL.PERS.	20	17	CAL PERS.	5	18	WVL PERS.	5	19	TOTL PERS.	65
20	CNSEC SCNS	5	21	LINEAR DEV	5	22	MNPCM SYNC	1020	23	A4 PCM END	956
24	MIN SAMPLS	610	25	MAX SAMPLS	685						
26	HST RNGE 1	38	27	HST RNGE 2	39	28	HST RNGE 3	40	29	HST RNGE 4	41
30	HST RNGE 5	42	31	HST RNGE 6	43	32	HST RNGE 7	44			
33	HST RNGE 1	2	34	HST RNGE 2	3	35	HST RNGE 3	4	36	HST RNGE 4	5
37	HST RNGE 5	6	38	HST RNGE 6	7	39	HST RNGE 7	8			
40	HST RNGE 1	9	41	HST RNGE 2	10	42	HST RNGE 3	11	43	HST RNGE 4	12
44	HST RNGE 5	13	45	HST RNGE 6	14	46	HST RNGE 7	15			
47	HST RNGE 1	3	48	HST RNGE 2	4	49	HST RNGE 3	5	50	HST RNGE 4	6
51	HST RNGE 5	7	52	HST RNGE 6	8	53	HST RNGE 7	9			
54	HST RNGE 1	37	55	HST RNGE 2	38	56	HST RNGE 3	39	57	HST RNGE 4	40
58	HST RNGE 5	41	59	HST RNGE 6	42	60	HST RNGE 7	43			
61	LWPCM PRAB	600	62	HIPCM PRAB	650						
63	MAX TOL 1	2	64	MIN TOL 1	0	65	MAX TOL 2	498	66	MIN TOL 2	494
67	MAX TOL 3	600	68	MIN TOL 3	0	69	MAX TOL 4	550	70	MIN TOL 4	0
71	MAX TOL 1	500	72	MIN TOL 1	0	73	MAX TOL 2	700	74	MIN TOL 2	0
75	MAX TOL 3	1023	76	MIN TOL 3	0	77	MAX TOL 4	550	78	MIN TOL 4	0
79	MAX TOL 1	900	80	MIN TOL 1	0	81	MAX TOL 2	650	82	MIN TOL 2	200
83	MAX TOL 3	185	84	MIN TOL 3	165	85	MAX TOL 4	535	86	MIN TOL 4	520
87	MAX TOL 1	2	88	MIN TOL 1	0						
89	MX FLD RAD	915	90	MN FLD RAD	865	91	MX HTD RAD	675	92	MN HTD RAD	625
93	MX SWL RAD	75	94	MN SWL RAD	25	95	MX AMB RAD	325	96	MN AMB RAD	275
97	MX BTH FOV	925	98	MN BTH FOV	875	99	MX SWL FOV	400	100	MN SWL FOV	350
101	MX LWL FOV	640	102	MN LWL FOV	520						

INDEX	PARAMETER	VALUE	INDEX	PARAMETER	VALUE	INDEX	PARAMETER	VALUE
1	WAVELENGTH	1.3300000	2	CAL SRC BR	8.72000027E-03	3	CHNL BIVLT	2.40000002E-02
4	RSP COEF 0	-478.10999	5	RSP COEF 1	385.12399	6	RSP COEF 2	0.00000000
7	RSP COEF 3	0.00000000						
8	WAVELENGTH	0.64999998	9	CAL SRC BR	6.00000005E-03	10	CHNL BIVLT	5.90000004E-02
11	RSP COEF 0	324.46499	12	RSP COEF 1	362.56201	13	RSP COEF 2	341.14401
14	RSP COEF 3	0.00000000						
15	WAVELENGTH	0.80000001	16	CAL SRC BR	9.25000012E-03	17	CHNL BIVLT	2.99999993E-02
18	RSP COEF 0	521.18799	19	RSP COEF 1	361.32800	20	RSP COEF 2	-27.342800
21	RSP COEF 3	0.00000000						
22	WAVELENGTH	12.300000	23	CAL SRC BR	1.00000000	24	CHNL BIVLT	0.20200001
25	RSP COEF 0	-545.40900	26	RSP COEF 1	60.747002	27	RSP COEF 2	[illegible].00000000
28	RSP COEF 3	0.00000000						

S 1 9 1 H P R E P R O C E S S O R T A P E Q U A L I T Y T E S T

START DATE: 6-MAY-76

RUN DATE: 24-JUN-76

*****ANOMALIES LIST*****

FRAME TIME	RECORD	PARAMETER TYPE NUMBER	PARAMETER	INDICATION
141 12: 10,0420	240	11	RAD CAL WHEEL POSITION	1,5250 VOLTS
141 13: 0,5690	421	11	RAD CAL WHEEL POSITION	118 ANOMALIES IN 30 SCANS
141 13: 0,5690	420	2	POWER SUPPLY DIAGNOSTICS	0,0000 VOLTS
141 13: 0,8495	421	10	AMBIENT CAL TEMPERATURE	11,7708 DEG, CENTIGRADE
141 13: 0,5690	420	11	RAD CAL WHEEL POSITION	0,8200 VOLTS
141 13: 0,5690	421	2	POWER SUPPLY DIAGNOSTICS	3 ANOMALIES IN 30 SCANS
141 13: 0,5690	421	10	AMBIENT CAL TEMPERATURE	9 ANOMALIES IN 30 SCANS
141 13: 0,5690	421	11	RAD CAL WHEEL POSITION	15 ANOMALIES IN 30 SCANS
141 15: 0,1500	632	11	RAD CAL WHEEL POSITION	3,2400 VOLTS
141 15: 30,0926	738	15	SWL CAL LAMP VOLTAGE	0,0050 VOLTS
141 15: 37,2961	766	18	SCAN REJECTED	590 SAMPLES IN A SCAN
141 15: 55,7310	831	11	RAD CAL WHEEL POSITION	672 ANOMALIES IN 30 SCANS
141 16: 19,3100	914	2	POWER SUPPLY DIAGNOSTICS	0,0000 VOLTS
141 16: 19,5907	915	10	AMBIENT CAL TEMPERATURE	11,5802 DEG, CENTIGRADE
141 16: 19,3100	915	2	POWER SUPPLY DIAGNOSTICS	3 ANOMALIES IN 30 SCANS
141 16: 19,3100	915	10	AMBIENT CAL TEMPERATURE	9 ANOMALIES IN 30 SCANS
141 16: 19,3100	915	15	SWL CAL LAMP VOLTAGE	1056 ANOMALIES IN 30 SCANS

S 1 9 1 H PREPROCESSOR TAPE QUALITY TEST

RUN DATE: 24-JUN-76

*** Q A SUMMARY ***

FIRST FRAME TIME:14: 9:26.1525   LAST FRAME TIME: 0: 0: 0.0000

MISSION 99 FLIGHT 1 SITE 2 LINE 3 RUN 4

PARAMETER ANOMALY TOTALS

PARAMETER	TOTAL DETECTED	PARAMETER	TOTAL DETECTED
SCAN LINEARITY	0.	SCAN REJECTED	1.
SYNC PULSE	0.	SWL CAL LAMP	1056.
ZERO VOLTS REF.	0.	POWER SUPPLY DIAG.	6.
PACKAGE TEMP.	0.	DATA PALLET TEMP.	0.
LWL DETECTOR TEMP.	0.	SPEC. PALLET TEMP.	0.
DICHROIC TEMP.	0.	HEATED CAL TEMP.	0.
REF. SOURCE TEMP.	0.	AMBIENT CAL TEMP.	18.
INT. SPHERE TEMP.	0.	MIRROR TEMP.	0.
RAD CAL WHEEL POS.	805.	FOV FLAG	0.

HISTORICAL FILE FLAGS

NO CALCULATIONS ABORTED

S191H PREPROCESSOR TAPE QUALITY TEST PAGE 2

RUN DATE: 24-JUN-76

*****HISTORICAL FILE*****

START DATE: 18-APR-76 START TIME: 15: 8:50, 0

STOP TIME: 15: 9: 0, 0

MISSION 99 FLIGHT 1 SITE 2 LINE 3 RUN 4

FIRST FRAME TIME: 15: 8:50, 417

BIAS VOLTAGE HISTOGRAM

CHANNEL	VALUE	<OR= 38	39	40	41	42	43	>OR= 44
A1		166	192	238	245	208	167	113

CHANNEL	VALUE	<OR= 2	3	4	5	6	7	>OR= 8
A2		10	78	422	585	211	20	3

CHANNEL	VALUE	<OR= 8	9	10	11	12	13	>OR= 14
A3		78	76	166	309	288	201	211

CHANNEL	VALUE	<OR= 3	4	5	6	7	8	>OR= 9
A5		0	0	5	1230	94	0	0

CHANNEL	VALUE	<OR= 37	38	39	40	41	42	>OR= 43
A6		133	141	190	208	170	188	299

S 1 9 1 H P R E P R O C E S S O R T A P E Q U A L I T Y T E S T

RUN DATE: 24-JUN-76

*****HISTORICAL FILE*****

START DATE: 18-APR-76

START TIME: 15:15: 5. 0
STOP TIME: 15:15:30. 0

MISSION 99 FLIGHT 1 SITE 2 LINE 3 RUN 4

FIRST FRAME TIME: 15:15: 4.2740

\-

WAVELENGTH CALIBRATION

CHANNEL A4 PCM VALUES AVG. = 620.6517 STD. DEV. = 0.6476 DATA FROM CH. A3

***** H I S T O R I C A L F I L E *****

START DATE: 15-APR-76 START TIME: 15:16: 0. 0
STOP TIME: 15:21: 0. 0

MISSION 99 FLIGHT 1 SITE 2 LINE 3 RUN 4
FIRST FRAME TIME: 15:16: 0.7429

CALIBRATION PERIOD

PARAMETER	VALUE	PARAMETER	VALUE
AVG. W/L RAMP PEAK VOLT	4,8679 VOLTS	SWL CAL LAMP	4,2249 VOLTS
W/L RAMP PEAK STD. DEV.	0,0044 VOLTS	RESPONSIVITY (A2)	153,6410 V/W-CM2-STR-MICR
AVG. W/L RAMP MIN. VOLT	0,0120 VOL S	NOISE (A2)	0,0382 VOLTS
W/L RAMP MIN. STD. DEV.	0,0027 VOLTS	NESR (A2)	2,4918 E-4 W/CM2-STR-MICR
AVG. LINEARITY OF SCAN	0,0019 RMS DEV	RESPONSIVITY (A3)	655,9166 V/W-CM2-STR-MICR
AVG. SCAN INTERVAL	0,9842 SECONDS	NOISE (A3)	0,3268 VOLTS
		NESR (A3)	4,9823 E-4 W/CM2-STR-MICR
		RESPONSIVITY (A5)	79,7837 V/W-CM2-STR-MICR
		NOISE (A5)	0,0112 VOLTS
		NESR (A5)	1,4086 E-4 W/CM2-STR-MICR
		RESPONSIVITY (A6)	1750,9989 V/W-CM2-STR-MICR
		NOISE (A6)	0,0545 VOLTS
		NESR (A6)	0,3117 E-4 W/CM2-STR-MICR

APPENDIX C

ERROR MESSAGES

11 CHANNELS REQUESTED DO NOT CORRESPOND TO ACTIVE CHANNELS ON FLIGHT TAPE
12 ILLEGAL CHANNEL RATIO REQUESTED, ONLY ONE CHANNEL REQUESTED
29 INVALID TIME REQUEST FOR OUTPUT PRODUCTS
43 INVALID SENSOR CODE INPUT TO FIELD OF VIEW PROCESSOR
49 EXCESSIVE SUBFRAME SYNC LOSS
55 REQUESTED SKYNET TIME NOT CONTAINED IN TAPE INTERVAL
69 JOB TERMINATED DUE TO CARD INPUT ERRORS
80 TOBCPR - ARGUMENT ERROR - TOCNTL
90 DISK FILE ALLOCATION REQUEST FOR TEMPORARY STORAGE OF TAPE HEADER BY SENSOR PROGRAM DENIED; REQUIRED FILE LENGTH EXCEEDS FREE SPACE AVAILABLE
151 ALL TIME SLICES PROCESSED
161 TIME SLICE TOO BIG - END OF PROCESSING
205 EXCESSIVE MAINFRAME SYNC LOSS
211 MEASUREMENT ID FOR CALIBRATION DATA NOT FOUND ON CALIBRATION FILE
219 END OF PHYSICAL SKYNET TAPE DETECTED
239 TAPE READ ERROR WHILE TRYING TO READ THE SKYNET TAPE
249 OUTPUT ARRAY IN CALCAT FOR COEFFICIENTS AND VALUES IS INSUFFICIENTLY DIMENSIONED
250 DCRIPT HAS DETECTED THAT THE BYTE ADDRESS OF THE BEGINNING OF THE DATA BLOCK IS ODD, DCOM2N EXPECTS AN EVEN BYTE ADDRESS (POSSIBLE ERROR IN NON-IMAGERY INPUT TAPE)
260 INVALID BASE DATE INPUT TO SKYNET
280 TAPE READ ERROR OCCURRED WHILE READING THE INPUT IMAGERY DATA TAPE, PROCESSING ABORTED
300 DCOM21 - THE OUTPUT BUFFER IS NOT LARGE ENOUGH TO CONTAIN THE HEADER RECORD OF THE INPUT IMAGERY TAPE.
310 DCOM21 - THE OUTPUT BUFFER IS NOT LARGE ENOUGH TO CONTAIN EITHER ONE SCAN LINE FOR ONE CHANNEL OF IMAGERY DATA , OR ONE ANCILLARY BLOCK WITH DOUBLE PRECISION TIME
320 DCOM21 HAS BEEN OVERLAYED BETWEEN CALLS, CANNOT RETRIEVE INFORMATION NEEDED TO COMPLETE TRANSMISSION OF DATA IN A DATA SET THAT WAS STARTED IN THE PREVIOUS CALL
330 THE TIME SLICE REQUESTED WAS LESS THAN THE FIRST TIME ENCOUNTERED ON THE INPUT IMAGERY DATA TAPE
340 NO SENSOR DATA OR ADAS DATA FOUND ON HIGH DENSITY TAPE IN THE TIME INTERVAL SPECIFIED
350 NO ADAS DATA FOUND ON THE HIGH DENSITY TAPE IN THE TIME INTERVAL SPECIFIED
360 NO SENSOR DATA FOUND ON THE HIGH DENSITY TAPE IN THE TIME INTERVAL SPECIFIED
370 DECOM1 SEARCHED 50 RECORDS AND COULD NOT FIND THE DIRECTORY ON THE HIGH DENSITY TAPE (POSSIBLE BAD HIGH DENSITY TAPE)

380 DECOM1 COULD NOT FIND TASK LISTED IN THE DIRECTORY THAT SATISFIED THE FORMAT AND TIME CONSTRAINTS, THE DESIRED DATA IS NOT ON THE HIGH DENSITY TAPE
390 DECOM1 DETECTED POS OUT OF LIMITS ON CALL TO THE DATA FILE DECODER (DFD). (POSSIBLE ERROR IN NON-IMAGERY INPUT TAPE)

400 DECOM1 DETECTED LCPF OR INCR OUT OF LIMITS ON CALL TO THE DATA FILE DECODER (DFD). (POSSIBLE ERROR IN NON-IMAGERY INPUT TAPE)
410 DECOM1 DETECTED LCPF OR FSTIND OUT OF LIMITS ON CALL TO THE DATA FILE DECODER(DFD). (POSSIBLE ERROR IN NON-IMAGERY INPUT TAPE)
420 DECOM1 REACHED END OF DISK BEFORE THE INDICATED NUMBER OF RECORDS IN THE DISK FILE WERE WRITTEN FROM THE HIGH DENSITY TAPE TO DISK
430 DEVICE ERROR ON DISK WHILE WRITING HIGH DENSITY TAPE DATA TO DISK
440 DECOM1 COULD NOT FIND TIME OR FORMAT IN THE DESCRIPTOR FILE, HIGH DENSITY TAPE IS FORMATTED INCORRECTLY
460 THE REQUESTED FORMAT AND THE FORMAT IN THE DESCRIPTOR OF THE NON-IMAGERY INPUT TAPE DO NOT MATCH
470 MEAS IS NOT DIMENSIONED LARGE ENOUGH TO CONTAIN THE COMMUNICATION BETWEEN DCRIPT AND DCOM2N
480 DCOM2N CANNOT RETURN A MULTIPLE OF NAV FRAMES BECAUSE OF THE BUFFER SIZE
490 THE FIRST RECORD ON THE NON-IMAGERY TAPE IS NOT THE DESCRIPTOR FILE
500 THE MEASUREMENT ID LIST REQUESTED DOES NOT MATCH ANY OF THOSE IN THE BLOCK COMMON/DCCNTL/, (THE NON-IMAGERY INPUT SOURCE IS POSSIBLY IN ERROR)
510 DCRIPT DETECTED THAT THE USER REQUESTED THE DESCRIPTOR BE RETURNED IN THE BUFFER, BUT THE BUFFER IS NOT LARGE ENOUGH TO CONTAIN IT
520 DEVICE ERROR WHILE READING NON-IMAGERY DATA FROM A 9-TRACK TAPE OR A DISK FILE
530 DCRIPT COULD NOT LOCATE EITHER THE FORMAT IDENTIFIER BLOCK OR TIME IDENTIFIER BLOCK IN THE DESCRIPTOR, THE DESCRIPTOR FILE OF THE NON-IMAGERY TAPE IS IN ERROR
540 DCOM2N DETECTS THAT THERE ARE MORE THAN 100 PRIME FRAMES OF DATA PER RECORD, DCOM2N WILL NOT HANDLE SUCH RECORDS, THE NON-IMAGERY INPUT SOURCE IS POSSIBLY IN ERROR
545 THE REQUESTED TIME IS NOT ON THE NON-IMAGERY DATA TAPE
550 DCOM2N COULD NOT LOCATE THE BEGINNING OF A DATA CYCLE FOR SUBCOMMUTATED DATA. (POSSIBLE ERROR IN THE NON- IMAGERY DATA TAPE)
560 DCRIPT CANNOT DETERMINE IF A REQUESTED MEASUREMENT IS MAIN FRAME, SUBCOMMUTATED, OR SUPERCOMMUTATED. (POSSIBLE ERROR IN THE NON-IMAGERY DATA TAPE)
570 DCRIPT OR DCOM2N DETECTS THE MAXIMUM NUMBER OF MEASUREMENTS REQUESTED FROM THE NON-IMAGERY DATA TAPE FOR TWO FORMATS EXCEEDS 40, WHICH IS THE MAXIMUM ALLOWABLE
580 DCOM2N DETECTED THAT POS WAS OUT OF RANGE FOR THE DATA FILE DECODER (DFD). (POSSIBLE ERROR IN NON-IMAGERY DATA TAPE.)

590 DCOM2N DETECTED THAT LCPF OR INCR WAS OUT OF RANGE FOR THE DATA FILE DECODER (DFD). (POSSIBLE ERROR IN NON-IMAGERY DATA TAPE)
600 DCOM2N DETECTED LCPF OR FSTIND WAS OUT OF RANGE FOR THE DATA FILE DECODER (DFD). (POSSIBLE ERROR IN NON-IMAGERY DATA TAPE)
610 A MEASUREMENT SPECIFIED COULD NOT BE FOUND ON THE NON-IMAGERY DATA TAPE. (POSSIBLE ERROR IN NON-IMAGERY DATA TAPE)

620 DCOM21 WILL NOT HANDLE PICTURE ELEMENTS REQUIRING MORE THAN 24 BITS
621 DCOM21 DETECTED DATA WAS ORDERED BY PIXEL RATHER THAN BY CHANNEL, CANNOT HANDLE THIS CASE
700 DEVICE ERROR ON DISK IN PROGRAM WHICH OUTPUTS UNIVERSAL FORMAT CCT
710 DEVICE ERROR ON TAPE WHILE ATTEMPTING TO OUTPUT UNIVERSAL FORMAT CCT
730 A MEASUREMENT IN THE REQUESTED LIST OF PARAMETERS, FROM THE NON-IMAGERY TAPE, REQUIRES MORE THAN 300 WORDS PER RECORD, THE DECOMMUTATION PROGRAM, DCOM2N, WILL NOT HANDLE SUCH RECORDS
801 DISK FILE ALLOCATION REQUEST BY SENSOR PROGRAM DENIED, REQUIRED FILE LENGTH EXCEEDS FREE SPACE AVAILABLE
802 DISK READ/WRITE ERROR ON TEMPORARY DISK FILE IN SENSOR PROGRAM
890 TAB DESCRIPTION REQUESTED NOT AVAILABLE TO TBSUPR
891 THE TAB ID ASSOCIATED WITH THE PRINT SUPPRESSION TOLERANCES IS NOT THE SAME AS THE TAB ID ON CARD TYPE 6
892 UNRECOVERABLE READ ERROR ON NON-IMAGERY INPUT TAPE
900 ARRAY FOR STORAGE OF PLOT DESCRIPTIONS IS TOO SMALL TO CONTAIN PLOT DESCRIPTIONS ENTERED BY LEADCARDS
910 TOO MANY DYNAMIC HEADING PARAMETERS ARE BEING ENTERED IN LEAD CARDS
920 ARRAY FOR STORAGE OF TAB DESCRIPTIONS IS TOO SMALL TO CONTAIN TAB DESCRIPTIONS ENTERED BY LEAD CARDS
999 INPUT START OR STOP TIME OUT OF SPECIFIED BOUNDS
1000 THE HIGH DENSITY TAPE HAS BEEN REWOUND
1000 PASS 1 COMPLETE
1050 DATA AVAILABLE WITHIN CURRENT TIME SLICE HAS BEEN PROCESSED. PROCEEDING TO PROCESS NEXT REQUESTED TIME SLICE
1075 SKYNET TELEMETRY DATA DROPOUT ENCOUNTERED
1131 ALL DATA PROCESSED
1141 CELL WITH CELL CENTER TIME DELETED
1149 SENSOR FIELD OF VIEW POINTING AXIS DOES NOT INTERSECT THE SURFACE OF THE EARTH
1150 NO TELEMETRY DATA AVAILABLE FOR FIELD OF VIEW COMPUTATIONS, NO VALUES RETURNED BY FIELD OF VIEW PROCESSOR
1181 I/O ERROR IN CALIBRATION SEQUENCE RESULTS - PROGRAM CONTINUES BUT DATA LOST
1191 SKIP IN DATA WITHIN TIME INTERVAL - INTERVAL CUT SHORT
1201 BAD DATA FOR THERMISTOR - DATA IN INTERVAL NOT USED
1211 CAL STEP NOT USED TO COMPUTE BASELINE COUNT
1221 CAL STEP NOT USED TO COMPUTE BASELINE TEMPERATURE
1231 CAL STEP NOT USED TO COMPUTE SCALE FACTOR
1291 CONTINUE PROCESSING
1260 READ ERROR ON NON-IMAGERY INPUT TAPE, A PHYSICAL RECORD WAS SKIPPED AND AN ATTEMPT IS BEING MADE TO PROCESS NEXT RECORD

1270 INVALID SKYNET ALTITUDE, VALUE LESS THAN OR EQUAL TO ZERO
1280 IMPROPER RANGE VALUE DOES NOT PERMIT SCATTEROMETER BACKSCATTER CALCULATION
1302 TOO MANY PLOT TYPES DESCRIBED IN LEAD CARDS, ONLY FIFTEEN ENTERED
1310 TOO MANY TAB TYPES DESCRIBED IN LEAD CARDS, ONLY FIVE ENTERED
803 START OF DATA CANNOT BE DETECTED
925 EBCDIC TO ASCII CONVERSION ERROR IN HEADER OF NON-IMAGERY INPUT TAPE
930 BAD FRAME SYNC ENCOUNTERED IN FIRST ANCILLARY OF MSS IMAGERY INPUT TAPE, PROCESSING STOPS
935 EXCESSIVE BAD SYNC CODES IN ADAS DATA
936 NO ADAS DATA IN MEAS ARRAY
937 T1 IS NOT WITHIN THE TIME LIMITS IN MEAS ARRAY
241 THE ADAS TIME ON THE INPUT TAPE CAUSED A DATA OVERFLOW. THIS OCCURS WHERE THERE IS AN ERROR IN THE ADAS DATA ON THE INPUT TAPE.
200 CHANNEL NEEDED FOR PRMSS COMPUTATIONS IS OUT OF SYNC-PROCESSING STOPPED
1300 MORE THAN ONE MAXIMUM MEAN VALUE IN VIDEO DATA, PROCESSING CONTINUES